

# Department of Veterans Affairs

## Software Design Document (SDD)

### Payer Operating Rules:

Final Claim Status (276/277) – CHAMPVA/SB/CWVV Claims

**VA**



**U.S. Department of Veterans Affairs**

Office of Information and Technology

*Product Development*

December 2013

Version 4.0

## Revision History

Date	Revision	Description	Author
12/3/12	0.1	Initial	██████
12/4/12	0.2	Initial review by tech writer	██████
12/6/12	0.3	Updated with routine design changes	██████
12/6/12	0.4	Final tech writer review	██████
12/10/12	0.5	Update version number	██████
12/10/12	0.6	Footer correction	██████
12/12/12	0.7	QA Review response	██████████████
12/13/12	1.0	Update version number	██████
1/17/13	1.1	Add further explanation to sections 2.1, 3.2.1.1, 3.2.2.1, and 3.2.2.2	██████
1/17/13	2.0	Changed version number	██████
12/4/13	3.0	Update proposed code for file size limit. Changed from Emdeon requirement of 500 mb to 150 mb because of transmission issues from HAC to Emdeon. Updated sections 3.2.1.4 and 3.2.2.3 to describe this file size change.	██████
12/11/13	4.0	Changed section 3.2.3.1 as a result of requested change to only include claims that have gone final since the release of the SLLA project software in production	██████

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Purpose .....	4
1.2	Scope.....	4
1.3	Definitions and Acronyms.....	4
1.3.1	Acronyms .....	4
1.3.2	Definitions .....	7
1.4	References .....	9
<b>2</b>	<b>Conceptual Design .....</b>	<b>9</b>
2.1	Overall Specifications .....	9
2.2	Accessibility Specifications.....	9
2.3	Disaster Recovery Specifications.....	9
2.4	Product Features .....	9
2.5	User Characteristics .....	9
2.6	Dependencies and Constraints .....	9
<b>3</b>	<b>Specific Requirements.....</b>	<b>10</b>
3.1	Database Repository .....	10
3.2	System Features .....	10
3.2.1	Final Claim Status – Primary Claims – CHAMPVA/SB/CWVV .....	10
3.2.1.1	Include Updated Claim Status for Primary Claims in the Final Claim Status file .....	10
3.2.1.2	Archive All Final Claim Status Files for Primary Claims.....	61
3.2.1.3	Transmit Daily Final Claim Status Files for Primary Claims.....	63
3.2.1.4	Create Final Claim Status File Data for Primary Claims .....	64
3.2.1.5	Add Line Item Control Number (LICN) to the Claim Line Level Detail Record for Primary Claims .....	93
3.2.1.6	Add LICN to the Claim Status Record Detail Record for Primary Claims.....	96
3.2.1.7	Check/EFT Date for Split Claim Submission for Primary Claims.....	98
3.2.1.8	Check/EFT Number for Split Claim Submission for Primary Claims.....	100
3.2.1.9	Claim Adjudication/Payment Date for Denied Split Claim for Primary Claims.....	103
3.2.1.10	Claim Adjudication/Payment Date for Paid Split Claim for Primary Claims.....	106
3.2.2	Final Claim Status – Secondary Claims – CHAMPVA.....	110
3.2.2.1	Include Updated Claim Status for Secondary Claims in the Final Claim Status File .....	110
3.2.2.2	Archive All Final Claim Status Files for Secondary Claims .....	136
3.2.2.3	Create Final Claim Status File Data for Secondary Claims .....	138
3.2.2.4	Add LICN to the Claim Line Level Detail Record for Secondary Claims.....	167
3.2.2.5	Add LICN to the Claim Status Detail Record for Secondary Claims.....	169
3.2.2.6	Check/EFT Date for Split Submission for Secondary Claims .....	171
3.2.2.7	Check/EFT Number for Split Submission for Secondary Claims .....	174
3.2.2.8	Claim Adjudication/Payment Date for Denied Split Claim for Secondary Claims .....	177
3.2.2.9	Claim Adjudication/Payment Date for Paid Split Claim for Secondary Claims.....	180
3.2.3	Final Status One-Time Update for Primary Claims for CHAMPVA/SB/CWVV .....	184
3.2.3.1	Create a Final Status Update for Historical Primary Claims .....	184
	<b>Attachment A - Approval Signatures.....</b>	<b>187</b>

# 1 Introduction

The PC@HAC employs the CP&E system to supply claim adjudication data to Emdeon. When a provider submits a X12N 5010 Health Claim Status Request (276) through the Emdeon clearinghouse network, Emdeon produces a X12N 5010 Health Claim Status Response (277) with the current claim status.

The 5010 software development project provided the ability to transmit Acknowledgment and Pending Claim Statuses for HAC EDI X12N 5010 Health Care Claim (837) to Emdeon. However, there is no process that transmits Final Claims Statuses to Emdeon. This Software Design Document will describe the design for implementing the capability to transmit the Final Claim Statuses to Emdeon.

## 1.1 Purpose

The Software Design Document (SDD) outlines the design that will support the requirements for the HAPE Payer Operating Rules (POR) initiative. Specifically, this activity involves the creation and the process of transmitting the claim and line level data required by Emdeon to create an ASC X12N 5010 Health Claim Status Response (277) for electronic CHAMPVA/SB/CWVV claims with a final claims status.

The target audience for this SDD includes the Chief Business Office (CBO), Office of Enterprise Development (OED), Harris development staff, Emdeon (the Health Care Clearinghouse [HCCH]), Software Quality Assurance (SQA), and the end-users.

## 1.2 Scope

The modifications to the Claim Status files only apply to the CHAMPVA, SB, and CWVV programs.

This effort will include changes to the following:

- Create, save and transmit a daily non-COBC Final Claim Status file for primary electronic CHAMPVA/SB/CWVV claims, with claim and line level details.
- Create and save a daily COBC Final Claim Status file for secondary electronic CHAMPVA claims, with claim and line level details.
- Upon implementation, transmit a file(s) to update the existing claim statuses for electronic primary claims with a final status that currently remain in a pending status for a predetermined amount of time.

## 1.3 Definitions and Acronyms

### 1.3.1 Acronyms

Acronyms/ Abbreviations/Terms	Description
ACA	Patient Protection and Affordable Care Act
AET	Automated Eligibility Tool
AI	Artificial Intelligence
AITC	Austin Information and Technology Center
ART	Adverse Reaction Tracking

<b>Acronyms/ Abbreviations/Terms</b>	<b>Description</b>
ASC X12	Accredited Standards Committee X12
BIRLS	Beneficiary Identification and Records Locator System
CAQH	Council for Affordable Quality Healthcare
CBO	Chief Business Office
CHAMPVA	Civilian Health and Medical Program for the Department of Veterans Affairs
CITI	CHAMPVA In-house Treatment Initiative
CMOP	Consolidated Mail Order Pharmacy
CMS	Centers for Medicare and Medicaid Services
CO	Contractual Obligation (Provider Responsibility)
COB	Coordination of Benefits
COBC	Coordination of Benefits Contractor
CORE®	Committee on Operating Rules for Information Exchange
CP&E	Claims Processing and Eligibility system
CPD	Claims Processing Division
CPS	Claims Processing System
CSR	Customer Service Representative
CWVV	Children of Women Vietnam Veterans
DEERS	Defense Enrollment Eligibility Reporting System
DME	Durable Medical Equipment
DOS	Date of Service
DoD	Department of Defense
DRG	Diagnosis-related Group
DX	Diagnosis
EAS	Enrollment Application System
EDI	Electronic Data Interchange
EEC	Eligibility and Enrollment Center
EFT	Electronic Fund Transfer
EIN	Employer Identification Number
EEOB	Electronic Explanation of Benefits
EOB	Explanation of Benefits
ESI	Eligibility and Enrollment System Improvements
ESR	Enrollment System Re-Design
EWV	EDI Web Viewer
FBCS	Fee Basis Claims System
FMS	Financial Management System

<b>Acronyms/ Abbreviations/Terms</b>	<b>Description</b>
FPPS	Fee Payment Processing System
FSC	Financial Services Center
FTEs	Full Time Equivalent
FTP	File Transport Protocol
FY	Fiscal Year
GUI	Graphical User Interface
HAC	Health Administration Center. The organization is now known as Purchased Care @ Health Administration Center (PC@HAC).
HEC	Health Eligibility Center
HECMS	Health Eligibility Case Management System
HCCH	Health Care Clearinghouse
HCPCS	Healthcare Common Procedure Coding System
HHS	Department of Health and Human Services
HIPAA	Health Insurance Portability and Accountability Act of 1996
HL7	Health Level 7
HPID	Health Plan Identifier
ICD-10	International Classification of Diseases, 10 <sup>th</sup> revision
ICD	Interface Control Document
ICN	Integrated Control Number
IP	Initial Processing
IT	Information Technology
LEIE	List of Excluded Individuals/Entities
LICN	Line Item Control Number
MbM	Meds by Mail
M (MUMPS)	Massachusetts General Hospital Utility Multi-Programming System
MIA	835 Segment used for Remittance Remark Codes on Inpatient Claims
MIPPA	Medicare Improvements for Patients and Providers Act
MOA	835 Segment used for Remittance Remark Codes on Non-Inpatient Claims
MPI	Master Patient Index
MRA	Master Record Adjustments
MSDS	Military Service Data Sharing
MVI	Master Veteran Index
NCPDP	National Council for Prescription Drug Programs
NDC	National Drug Code
NFPO	National Fee Program Office

Acronyms/ Abbreviations/Terms	Description
NUBC	National Uniform Billing Committee
PC@HAC	Purchased Care @ Health Administration Center

### 1.3.2 Definitions

Term	Definition
270/271 Health Care Eligibility Benefit Inquiry and Response	A transaction set for Eligibility and Benefits, used to inquire about and receive information about eligibility/benefits for a specific patient from a payer.
276/277 Health Care Claim Status Request and Response	A transaction set for health care claims, used to inquire about and receive information about the processing status of a claim.
837 Health Care Claim	A transaction set for medical and dental health care claims, used to send a claim to a trading partner or payer.
835 Health Care Claim Payment/Advice	A transaction set for health care claim payment advice (or remittance advice) - referred to as a Remittance advice.
999 Implementation Acknowledgement	A transaction set for Health Care Claim acknowledgements, used to report a syntactical and relational analysis or to acknowledge receipt of an error-free transaction set.
AI	Artificial Intelligence - rules-based system used to validate compliance of health care claims to the Health Administration Center's policies.
ASC X12	The Accredited Standards Committee X12 – is an ANSI-accredited standards development organization and the entity responsible for the HIPAA transaction standards for electronic health care, eligibility, claims processing, claims status, authorizations and remittance transactions named by the Health Insurance Portability and Accountability Act of 1996.
Caché	MUMPS/CACHÉ, an application and database environment (InterSystems).
CAQH	Council for Affordable Quality Healthcare (CAQH) is a nonprofit alliance of health plans and trade associations whose goal is to simplify healthcare administration through industry initiatives.
CMS-1500	The National Uniform Claim Committee (NUCC) paper health insurance claim form for professional claims prescribed by many health plans for claims submitted by physicians, suppliers, and ambulance services.
COBC	The CMS Coordination of Benefits Contractor (COBC) routes COB Medicare Part A and B claims from CMS to other payers (i.e., CHAMPVA). The current COBC is Group Health Incorporated (GHI).
CORE®	The Committee on Operating Rules for Information Exchange (CORE®) is a multi-stakeholder initiative created, organized and facilitated by CAQH that is working to make it easier for physicians and hospitals to access eligibility, benefits and claim information for their patients at the point of care.
EFT	Electronic funds transfer is the process used to make claim payments directly to a provider's bank.

Term	Definition
Emdeon®	The contracted Purchase Care clearinghouse that provides a network of providers for the routing of electronic Institutional/Professional and Dental health care claims (837) to the PC@HAC and delivery of the electronic remittance transactions (835). It also routes electronic coordination of benefits claims from the COBC to the PC@HAC for processing and provides support for the real-time processing of eligibility and benefits requests/responses.
EOB	Explanation of Benefits (EOB) is usually a paper health care claim payment/advice from payers that provides data pertaining to the claim adjudication and the amounts paid by the payer.
HIPAA v. 5010	HIPAA version 5010 is the newest set of standards related to the electronic transmission of specific health care transactions such as Health Care Claims, Eligibility Inquiry/Response, and Health Care Claim Remittance Advice.
HL7	Health Level Seven (HL7) is a non-profit standards organization that provides standards for interoperability that improve care delivery, optimize workflow, reduce ambiguity, and enhance knowledge transfer among stakeholders. "Level Seven" refers to the seventh level of the International Organization for Standardization (ISO) seven-layer communications model for Open Systems Interconnection.
NCPDP	The National Council for Prescription Drug Programs (NCPDP) is a not-for-profit ANSI-accredited standards development organization representing the pharmacy services industry and the entity responsible for development of standards for pharmacy, including real-time claims processing and e-prescribing.
NCPDP D.0	HIPAA version D.0 is the newest set of standards related to the electronic transmission of pharmacy transactions named by HIPAA legislation for the pharmacy industry.
NCPDP – B1	An NCPDP transaction set for billing a payer for a medication or medical item from a pharmacy.
NCPDP – B2	An NCPDP transaction set for the reversal of a previously submitted NCPDP – B1.
Payer	An insurance company, fiscal intermediary, government agency, other agency, or individual responsible for the payment of health care claims.
SXC	SXC Health Solutions Inc. is the contracted Pharmacy Benefit Manager (PBM) that provides a network of retail pharmacies for Purchased Care beneficiary programs and provides connectivity to Medicare Part D plans and VAMC pharmacies.
UB-04	The National Uniform Billing Committee (NUBC) paper health insurance claim form for institutional claims prescribed by many health plans for claims submitted by hospitals, nursing homes, hospices, home health agencies and other institutional providers.
User	The person or persons who operate or interact directly with the various systems which support the National Fee Program or the Health Administration Center.



## 1.4 References

Name	Location	Date
CORE Phase I Rules	<a href="http://[REDACTED]">http://[REDACTED]</a>	
CORE Phase II Rules	<a href="http://[REDACTED]">http://[REDACTED]</a>	
Emdeon®/Hosted Real-Time™ Claim Status Implementation Specification Version 03	<a href="http://[REDACTED]">http://[REDACTED]</a>	October 2010

## 2 Conceptual Design

### 2.1 Overall Specifications

The CP&E System currently creates the Acknowledgment and Pending Claim Status files for transmission to Emdeon. The FTP process transmits these files on a daily basis. This project will add the Final Claim Status files to the transmissions to Emdeon. The criteria will be that a Final Status will be generated when all of the claims, split-claims, and line items for an Original submission are either paid or declined.

### 2.2 Accessibility Specifications

A TaskMan entry submits a job in the background which creates the Acknowledgment and Pending Claim Status files. A similar process will also generate the Final Claim Status files. No user interface is applicable.

### 2.3 Disaster Recovery Specifications

There are no disaster recovery requirements specific to this development effort. The affected routine is an integrated part of the CP&E system that exists at the HAC site. The normal backup and recovery procedures will incorporate the new files.

### 2.4 Product Features

- Implement the creation and the transmitting of the Final Claim Status Files defined by the HCCH and their Claim Status Implementation Specification.
- Modify the existing routine for the Claim Status file as defined by the Requirements Document.

### 2.5 User Characteristics

Each night, a background job creates the Acknowledgement and Pending Claim Status files. An additional background job will create the Final Claim Status files. No user interaction is necessary.

### 2.6 Dependencies and Constraints

This project depends on the completion of cycle 5 UAT Testing of the Line Level Service Adjudication project and C-35 project.

HCCH (Emdeon) will provide resources to support both integration and site testing for this software development effort.

The CP&E system will save and archive the new Final Claim Status files by the same processes that save and archive the existing Claim Status files.

## 3 Specific Requirements

### 3.1 Database Repository

The CP&E database (M globals) contains the data used to create the Claim Status Files. An FTP process transmits these files to the HCCH and then archives them in a post office.

### 3.2 System Features

#### 3.2.1 Final Claim Status – Primary Claims – CHAMPVA/SB/CWVV

##### 3.2.1.1 Include Updated Claim Status for Primary Claims in the Final Claim Status file

##### Functional Requirement: 2.8.1.1 Update Claim Status – Primary Claims

The CP&E system shall update an electronic primary CHAMPVA/SB/CWVV claim status to one of the following final claim statuses when received from FMS:

- F1 Finalized/Payment – VA has paid the claim/line
- F2 Finalized/Denial – VA has denied the claim/line

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.1.1	
Related Options		
<b>Related Routines</b>	<b>Routines “Called By”</b>	<b>Routines “Called”</b>
	<b>TaskMan process</b>	
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		

Routine Name	CHMXWB21
<pre> CHMXWB21 ;HAC/DLB;WEB 5010 277 UNSOLICITED STATUS/DRIVER PART 1;10/25/10 ;;1.0;CHAMPVA SYSTEM;;JULY 11,2011;Build 1 ;;V1.0; ;;CALLED BY ^CHMXG001 TO PERFORM THE "ACK" STATUS ;;CALLED BY TASKMAN TO PERFORM THE "PENDING" AND "FINAL" STATUS ;;CALLS ENTRY POINTS TO CHMXWB22,CHMXWBUT ;;HR-COB-Medicare-A/B-Begin-9372 (27-May-2010) ** ** ** 10/18/2011 ADDED TSTBLDCLM() ENTRY POINT TO VALIDATE THE REJECT VALUES BEING ** REPORTED IN THE CSTAT FILE ** 10/20/2011 MODIFIED THE LINE() FUNCTION TO FIX INCORRECT REJECT STATUS ** REPORTING IN THE CSTAT FILE ** 10/21/2011 ADDED THE SIX DATE GATHERING FUNCTIONS TO SUPPORT THE RULE ** PROVIDED BY BUSINESS GROUP FOR I/P/D CLAIMS ** 10/21/2011 ADDED THE GETCLMTYP() FUNCTION TO DETERMINE THE I/P/D CLAIM TYPE ** FOR BOTH 4010 AND 5010 INCOMING CLAIMS ** 10/24/2011 CHANGED THE DATERANGE FUNCTION TO A EXISTING DATE FUNCTION ** IN THE UTILITY ROUTINE: CHMXWBUT.INT ** 10/25/2011 CHANGED THE RENDERING PHYSICIAN NPI RETRIEVAL TO GET THE DATA ** FROM THE CHMXCLE(I,64) TIN DATA FIELD. ** 10/25/2011 ADDED A TEST FOR THE "U" VALUE RECEIVED FROM EMDEON TO RETURN "" ** IF NOT A "M" OR "F" (ONLY ACCEPTED VALUES) ** 10/26/2011 REMOVED ERROR MESSAGE OUTPUT FOR ERROR TRAP HANDLER ** ** ** 10/26/2011 CHANGED THE INSTITUTIONAL AND DENTAL START/END DATES FOR CLAIM ** LEVEL REPORTING TO GET START FROM ^CHMXCLE(I,1),FIELD 1 AND END ** TO ^CHMXCLE(I,1) FIELD 2 ** 11/4/2011 CHANGED THE DATE CHECK ROUTINE GETCLMI() TO RETURN THE CORRECT ** DATE FOR THE DESIRED FLAG ** ; ;1.0 IS THE INITIAL VERSION (HIPAA Ready LLC) ;2.0 UPDATES TO THE STATUS REPORT PER HIPAA II SPECIFICATIONS. ; *** Filenames should not be hard coded in files. See ^CHMX277 TRACKING ; GLOBAL FILE for the destination directory/filenames for the 5010 status ; files.        Q       .....       ; PRODUCTION ENTRY POINTS TO PROCESS THE CURRENT STATUS. ;       .....  EPACK(CHMXI) ; SET UP THE FILE/BATCH INDEX (PROVIDED BY CHMXG001)       ; CHMXI I INDEX INTO ^CHMXCL()       N COUNT,DIRFILE,GRPCNT,FILECNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR,CNT       N MAXCNT,EICNT,CHA,CHMEDCOB       S FILECNT=0,RUNTYPE="A",GRPCNT=1,EICNT=0       S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER       D INIT       S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA=""       S CHTPI=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:CHTPI="" ; GET CLAIM T/P FOR FILE BEING CHECKED       S CHTPI=\$O(^CHMXTP("C",CHTPI,"")) Q:CHTPI="" D ; GET "LOCAL" TP ID       S IDX=\$O(^CHMX277("B",CHTPI,0)) </pre>	

Routine Name	CHMXWB21
S:\$P(^CHMX277(IDX,0),"^",4) ["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETCHCLEI(CHMXI,CHTPI) ; RETRIEVE ^CHMXCLE INDEX Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPPENDTST N TEST S TEST=1 D EPPEND Q  EPPEND N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR N CHNODEDEF,BREF,CHIIKVAL,DRDATA,TRKDATA,MAXCNT,CHMEDCOB,IDX ; FILEMAN UPDATE VARIABLES N CHTO,CHFROM,TRKI,EICNT S RUNTYPE="P",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER .S IDX=\$O(^CHMX277("B",CHTPI,0)) .S:\$P(^CHMX277(IDX,0),"^",4) ["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .K CHTPARR .S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0) .S CHTPARR(CHTPI)="" ; SET CHTPARR UP FOR GETPENDEI .D GETTOFROM(CHTPI) Q:(CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPFINAL N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR N CHNODEDEF,BREF,CHIIKVAL,DRDATA,TRKDATA,MAXCNT ; FILEMAN UPDATE VARIABLES N CHTO,CHFROM,EICNT S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S MTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF	

Routine Name	CHMXWB21
	<pre> .K CHTPARR S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0) .S CHTPARR(CHTPI)=" .D GETTOFROM(CHTPI) Q:('CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  ***** ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; *****  EPREPTST N TEST S TEST=1 D EPREPLACEMENT Q  EPREPLACEMENT N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT,CHMEDCOB,IDX,CHTPX S RUNTYPE="H",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="REPLACEMENT" S TP=\$\$LISTTP^CHMXWB22 ; SELECT TRADING PARTNER S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER S IDX=\$O(^CHMX277("B",TP,0)) S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHXI="" .F S CHXI=\$O(^CHMXCL("B",IDATE,CHXI)) Q:CHXI="" D ..S CHMXI=CHXI ..I \$D(TEST) W !,"CHECKING:",\$\$DTCVRT(\$E(IDATE,1,7)) ; SHOW FILE DATES BEING PROCESSED ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S CHTPX=\$P(\$G(^CHMXCLA(CHX,1)),"^",1) Q:CHTPX="" ; GET CLAIM T/P FOR FILE BEING CHECKED ..S CHTPX=\$O(^CHMXTP("C",CHTPX,"")) ; CHECK SELECTED VS CLAIM TP ID ..I TP=CHTPX D ; FILE MATCHES THE SELECTED TP ...S CHTPI=TP ; CHTPI IS USED DOWNSTREAM ...I \$D(TEST) W " PROCESSING:",\$\$DTCVRT(\$E(IDATE,1,7)) ...I \$D(TEST) W !," EPREPLACEMENT: CHMXI= ",CHMXI," CHA= ",CHA," CHTPI= ",CHTPI," TP= ",TP ..S CHTPARR(TP)=" D GETCHCLEI(CHMXI,CHTPI) ; IF TP VALID, PROCESS FILE </pre>

Routine Name	CHMXWB21
<pre> I \$D(TEST) W !,"REPLACE: \$J= ",\$J," REC COUNT= ",\$G(^ZSC(\$J,0))," CHTPI= ",CHTPI," TP= ",TP Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  DTCVRT( DATE ) ; CONVERT DATE TO EXTERNAL FORMAT N EXTDATE S EXTDATE=\$E( DATE,4,5 )_"_"\$E( DATE,6,7 )_"_"(\$E( DATE,1,3)+1700) Q EXTDATE  ***** ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; *****  EPOVERTST N TEST S TEST=1 D EPOVERLAY Q  EPOVERLAY N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT S RUNTYPE="O",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="OVERLAY" S TP=\$\$LISTTP^CHMXWB22 D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHMXI="" .F S CHMXI=\$O(^CHMXCL("B",IDATE,CHMXI)) Q:CHMXI="" D ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S TPTIN=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:TPTIN="" ..I TP=\$O(^CHMXTP("C",TPTIN,"")) S CHTPARR(TP)=" D GETCHCLEI(CHMXI,TP) Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  ***** ; INIT PERFORMS LOCAL VARIABLE INITIALIZATION REQUIREMENTS FOR THE STATUS ; ; PROCESSES. ; *****  INIT ; K ^ZSC(\$J) S ^ZSC(\$J,0)=0 K TMP S GROUPID="",CHTPI=0 I '\$D( DATESTAMP ) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% </pre>	

Routine Name	CHMXWB21
<pre> .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND Q  ..... ; EPSTAT IS THE ENTRY POINT FOR THE ON-DEMAND REPORT FOR STATISICS ; .....  EPSTAT D ONDEMAND^CHMXWB22 Q  EPREGEN D REGEN^CHMXWB22 Q  ..... ; FILE CREATION FUNCTIONS ; .....  CREATEFILE S MAXCNT=500000 D FILELOOP Q  FILELOOP ; BUILD CLAIMS FROM ^ZSC(\$J) ARRAY (^CHMXCLE(I) VALUES) N CHCLEI,PDI D GETGRPCNT ; GET ESTIMATED GROUP FILE COUNT Q:GRPCNT=0 D GETFILE U DIRFILE ; CREATE ORIGINAL FILE D BLDHDR ; BLDHDR SETS COUNT=2 (SEE EMDEON SPEC) S CHCLEI=0 F S CHCLEI=\$O(^ZSC(\$J,CHTPI,CHCLEI)) Q:CHCLEI="" D .S EICNT=EICNT+1 .S STVAL=\$G(^ZSC(\$J,CHTPI,CHCLEI)) .D BLDCLM(CHCLEI,STVAL) ; GENERATE CLAIM/STC,DTL/STC RECORDS .D CHKSEQ ; CHECK FOR 500MB LIMIT, CREATE NEW FILE D BLDTRL ; BUILD THE TRAILER RECORD D CLOSEFILE ; CLOSE CURRENT FILE Q  GETGRPCNT ; ESTIMATE THE TOTAL NUMBER OF FILES REQUIRED N RECCNT,X S RECCNT=0 S X="" F S X=\$O(^ZSC(\$J,CHTPI,X)) Q:X="" S RECCNT=RECCNT+1 </pre>	

Routine Name	CHMXWB21
<pre> S GRPCNT=RECCNT\MAXCNT                                ; INTEGER DIVISION OF RECORD COUNT S:RECCNT#MAXCNT&gt;0 GRPCNT=GRPCNT+1 ;W !,"GRPCNT= ",GRPCNT Q  CHKSEQ                                ; HANDLE THE FILE SEQUENCE NUMBERING REQUIREMENTS I EICNT&gt;MAXCNT D .D BLDTRL                                ; OUTPUT TRAILER RECORD .D CLOSEFILE .D GETFILE U DIRFILE .D BLDHDR                                ; BLDHDR RESETS "COUNT" VARIABLE Q  FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN Q:"APF"[RUNTYPE                                ; A=ACK, P=PENDING, F=FINAL N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^03="_DIRFILE_"^04="_COUN T_"^05="_CHMXI S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20,"TRKDATA=".02=1^03="_DIRFILE_"^04="_CHTO _"^05="_CHFROM_"^06="_COUNT S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30,"TRKDATA=".02=1^03="_DIRFILE_"^04="_CHTO _"^05="_CHFROM_"^06="_COUNT S BREF=FMDATE S CHIKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT D ADD^CHHRLIBFM(CHNODEDEF,CHIKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL Q  ..... ;TEST ENTRY POINT TO TEST THE BLDCLM FUNCTION FROM A PDI VALUE ; THERE ARE MULTIPLE CROSS REFERENCES (SOME DOCUMENTED, SOME UNDOCUMENTED ; THAT ARE UTILIZED IN THE CLAIM PROCESSING SYSTEM. THE CVTPDI AND TSTBLDCLM ; FUNCTIONS UTILIZE THE MAJORITY OF THESE CROSS REFERENCES. .....  CVTPDI(CHREFN) N CHPCN,CHMXCLI,CHMXID,CHCLFI,SDATE,EDATE,CHCLFI,CHCLEI,IDX I '\$D(^CHMXCLE("PDI",CHREFN)) W !,"NO ^CHMXCLE("PDI",,CHREFN,) XREF" Q S CHPCN="" S1 S CHPCN=\$O(^CHMXCLE("PDI",CHREFN,CHPCN)) I CHPCN="" Q ;VALID PDI,NO DATA S CHMXCLI=0 S2 S CHMXCLI=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI)) I 'CHMXCLI Q ;VALID PDI, NO DATA S CHMXID="" S3 S CHMXID=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI,CHMXID)) I CHMXID="" Q S CHCLEI=\$P(CHMXID,"*",4) W !,"CLM SDATE [^CHMXCLE(I,1),1]=",\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) W !,"CLM EDATE [^CHMXCLE(I,1),2]=",\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) S CHCLFI=0 W !,"LINE ITEM DATE VALUES" </pre>	



Routine Name	CHMXWB21
	<pre> F IDX=1:1 S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:'CHCLFI D ; CHECK ALL LINE ITEM DATES .W !,IDX," ", " L/I SDATE [^CHMXCLF(I,1),11]=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) .W !," L/I END [^CHMXCLF(I,1),12]=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) .W !," THERAPY SDATE [^CHMXCLF(I,2.5),1]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),"^",1) .W !," LAST CERT DATE [^CHMXCLF(I,2.5),2]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),"^",2) Q  TSTBLDCLM(CHPDI) N CHPCN,CHCLI,CHCLAI,CHCLEI,STVAL,COUNT,PDI,BUFI,RUNTYPE,CLMTYPE,TEST S (STVAL,CHPCN)="", (COUNT,CHCLEI,CHCLAI,PDI)=0,TEST=1 S CHPCN=0,CHPCN=\$O(^CHMXCLE("PDI",CHPDI,CHPCN)) I CHPCN="" W !,"NO CHPCN VALUE" Q E W !,"PATIENT CONTROL # [^CHMXCLE("PDI",CHPDI,PCN)]= ",CHPCN ; PATIENT CONTROL NUMBER S CHCLI=0,CHCLI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI)) I CHCLI="" W !,"NO CHCLI VALUE" Q S STVAL=0,STVAL=\$O(^CHMXCLE("A",CHCLI,STVAL)) W !,"XREF CLAIM STATUS= ",STVAL S BUFI="",BUFI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI)) I BUFI="" W !,"NO BUFFER INDEX VALUES" Q W !,"XREF [\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))] BUFFER INDICES= ",BUFI S CHCLEI=\$P(BUFI,"*",4) I CHCLEI="" W !,"NO CHCLEI VALUE RECOVERED" S RUNTYPE="H" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; RETRIEVE PDI S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) W !,"PDI VALUE FROM [^CHMXCLE(I,100),2 OR 4]= ",PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) W !!,"X12 837 BUFFER [CHMXCL(I)= ",?30,CHCLI W !,"TRANSACTION [CHMXCLA(I)]= ",?30,CHCLAI W !,"PROVIDER BUFFER [CHMXCLB(I)]= ",?30,CHCLBI W !,"PATIENT BUFFER [CHMXCLC(I)]= ",?30,CHCLCI W !,"CLM BUFFER [CHMXCLE(I)= ",?30,CHCLEI W !,"L/I BUFFER [CHMXCLF(I)]= ",?30,\$O(^CHMXCLF("B",CHCLEI,0)) S CLMTYPE=\$\$GETCLMTYP W !,"VERSION-FUNCTIONAL TYPE [^CHMXCLA(I,0),13]=\$ ", \$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13) W !!,"CLAIM TYPE= ", \$\$S(CLMTYPE="I": "INSTITUTIONAL",CLMTYPE="P": "PROFESSIONAL",CLMTYPE="D": "DENTAL") ; I/P/D CLAIM W !,"STATUS (ACK/PENDING)=", \$\$S(\$D(^CHMPAY("C",PDI))=0:"ACK",1:"PENDING") D CVTPDI(CHPDI) W !! D BLDCLM(CHCLEI,"") Q  ..... ; BLDCLM: ENTRY POINT TO BUILD THE CSTAT RECORDS ; THIS FUNCTION PROVIDES AN EXTERNAL ENTRY POINT TO GENERATE THE ; EMDEON CLAIM,CLAIM STATUS,DETAIL,AND DETAIL STATUS RECORDS ..... </pre>

Routine Name	CHMXWB21
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP I '\$D(ATESTAMP) D                                ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D Q ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)=" " S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI=" " D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI=" " PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL() ; FINAL STATUS IMPLEMENTED LATER Q 0  ***** ; FUNCTIONS TO GATHER INFORMATION REGARDING REJECT REASONS ; ***** </pre>	

Routine Name	CHMXWB21
	<pre> BLDACK                                ; POPULATE CHRJARR REJECT REASON/REJECT CODE VALUES     K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY     S REJCODES=""     D GTRXRJRSN^CHMXWBUT(CHCLAI) ; TRANSACTION "101" NODE REJECT REASONS     D GPRORJRSN^CHMXWBUT(CHCLBI) ; PROVIDER "101" NODE REJECT REASONS     D GPATRJRSN^CHMXWBUT(CHCLCI) ; PATIENT "101" NODE REJECT REASONS     D GCLMRJRSN^CHMXWBUT(CHCLEI) ; CLAIM "101" NODE REJECT REASONS     I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*"     ;I \$D(TEST) W !,"BLDACK: RJCNT=",CHRJARR(0)," RJARR(1,1)=",CHRJARR(1,1)     Q  ..... ; RECORD BUILDING FUNCTIONS: \$TEXT FILES PROVIDE FIELD DEFINITIONS ; .....  BLDHDR                                ;BUILD HEADER RECORD     N LN,REC,STR,LOADTYPE     S (STR,LN,REC)="",COUNT=2 ; EMDEON SPEC: REC. COUNT STARTS @ 2     D GETHDRVAL                        ; SET UP HEADER VARIABLES     F LN=1:1 S STR=\$T(EMDEONHDR+LN) Q:STR["END OF RECORD" D     .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR)     .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR)     W REC,! S REC=""     Q  CLM     N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID     N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID,S UBLNAME     N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATNA MEX,PATDOB     N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM     N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT     N PDI,CHMPAYI,REC,LN,STR     I (RUNTYPE="F") D                                ; DO ACK AND PENDING     .S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, MUST BE "PENDING"     .S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4)     .S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0))     .D GETCLMVAL                                    ; SET UP THE CLAIM VALUE VARIABLES     .S REC=""  ; BUILD THE RECORD     .F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D     ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR)     ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR)     .W REC,! S REC="",COUNT=COUNT+1     Q  LINE                                ; BUILD DTL RECORDS FOR LINE ITEMS     N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE </pre>

Routine Name	CHMXWB21
	<pre> N CHJVAL,RJRSN,REC,STR,LN S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D .D GDTLVAR\$ .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" .D BLDSTC("DTL") Q  BLDSTC(TYPE) ; WRITE OUT THE STC RECORDS FROM CHRJARR REJECT ARRAY ; TYPE CLAIM OR DETAIL LEVEL EMDEON STATUS VALUES &amp; LINE ITEM CONTROL NUMBER N PCINUM,LICNUM,DATERR,ESCODE N IDX,JDX,REJCODES,ECNT N LN,STR,REC S REJCODES=0 S ECNT=CHRJARR(0) ; FRONT END EDIT REJECT COUNT F IDX=1:1 Q:(IDX&gt;ECNT) D ; PROVIDE STC RECORD FOR EACH ERROR .F JDX=1:1 Q:\$G(CHRJARR(IDX,JDX))="" D ; GET EACH REJECT VALUE STORED ..S (RJCODE,RJSTATUS,ENTITY)="" ..D GETSTCVAL(TYPE) ; EMDEON STATUS RECORD VALUES FOR CLM OR DTL ..S REJCODES=\$G(CHRJARR(IDX,JDX)) ..W:\$D(TEST) !,"BLDSTC: REJCODE STR=",REJCODES,! ; DEBUG: ONLY OUTPUTS IF "TEST" IS DEFINED ..S RJCODE=\$P(REJCODES,"*",1),RJSTATUS=\$P(REJCODES,"*",2),ENTITY=\$P(REJCODES, "*,3) ..S REC="" ..F LN=1:1 S STR=\$T(EMDEONSTC+LN) Q:STR["END OF RECORD" D ; GENERATE THE STATUS (STC) RECORD ...I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ...E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 Q  BLDTRL ; TRAILER RECORD N LN,REC,STR S COUNT=COUNT-2 ; ADJUST COUNT FOR START OF 2 (SEE EMDEON SPEC) S (STR,LN,REC)="" F LN=1:1 S STR=\$T(EMDEONTRLR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ; 10/20/2010 ADDED "ELSE" W REC,! S REC="" Q  ..... ; GENERIC FUNCTIONS SUPPORTING THE RECORD BUILDING ; ..... </pre>

Routine Name	CHMXWB21
<pre> GETFILE()   N DIRPATH   S EICNT=0,FILECNT=FILECNT+1   S DIRPATH="CHAMPVA_EDI:[KERMIT.WEBMD]"   HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP   D NOW^%DTC   S FMDATE=%   S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")   S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")   S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND   I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT"   E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT" ; CHMEDCOB SET IN CHMXWT1   E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME   I RUNTYPE="H" S DIRFILE=DIRFILE_"_H"   I RUNTYPE="O" S DIRFILE=DIRFILE_"_O"   S FLAG=\$\$OFILE^CHMXWBUT(DIRFILE,"NWS")   I 'FLAG D VMSERR Q ""   S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME   ;E W !,"OPENED ",DIRFILE,!   Q  TST   N FMDATE,TIME   S FMDATE=3110901.1234   W !,FMDATE   S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")   W !,FMDATE   S TIME=\$E(FMDATE,9,14)   W !,TIME   S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")   W !,TIME   Q  CLOSEFILE   C DIRFILE   Q  GETCHCLEI(CHCLI,CTPI) ; BUILD ARRAY OF CLAIM INDICES (CHMXCLE(I) VALUES) ; CHCLI I INDEX FOR ^CHMXCL() FILE (CLAIM FILE RECEIVED) N STAT,CHCLAI,PDI,CHIDXS,CHCLEI,RECCNT,CTPI S CTPI=0 S STAT="",STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) I \$D(TEST) W:STAT="" !,"NULL STAT" Q:STAT="" ; RETRIEVE STATUS VALUE S CHCLAI="",CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) I \$D(TEST) W:CHCLAI="" !,"NULL CHCLAI" Q:CHCLAI="" ; RETRIEVE THE ^CHMXCLA(I) VALUE I \$D(TEST) W !," GETCHCLEI: READING:",CHCLI," STATUS= ",STAT," ^CHMXCLA(I)= ",CHCLAI," TP= ",CTPI S STAT="" F S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) Q:STAT="" D .S CHCLAI="" .F S CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) Q:CHCLAI="" D ..S PDI=0 ..F S PDI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI)) Q:'PDI D </pre>	

Routine Name	CHMXWB21
<pre> ...S CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,0)) ...Q:CHIDXS="" ; PDI RETRIEVES BUFFER IDXs ...S CHCLEI=\$P(CHIDXS,"*",3) ...S ^ZSC(\$J,CHTPI,CHCLEI)=STAT,RECCNT=\$I(^ZSC(\$J,0)) Q  GETTP(CHA) N TPID,ID S ID="" S TPID=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) S:TPID]"" ID=\$O(^CHMXTP("C",TPID,"")) Q ID  GETPENDEI(CHFROM,CHTO) ; GET CHCLEI FROM PENDING ^CHMPAY(I) FROM/TO VALUES K ^ZSC(\$J) N IDX,CHJK,PDI,CHCLEI,CHCLAI,CTPI,RECCNT S RECCNT=0 F IDX=CHFROM:1 Q:IDX&gt;CHTO D .S CHJK=\$O(^CHMPAY(IDX,"PDI",0)) Q:CHJK="" .S PDI=\$P(^CHMPAY(IDX,"PDI",CHJK,0),"^",1) Q:PDI="" .S CHCLEI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",6) Q:CHCLEI="" .S CHCLAI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",3) Q:CHCLAI="" .S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM .I CHCLEI]"" I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1 S ^ZSC(\$J,0)=RECCNT Q  ; ;..... ; GETCHTPI RETURN THE NEXT TRADING PARTNER INDEX BASED ON THE LAST INDEX ; ; USED. THIS FUNCTRION IS CALLED BY THE "PENDING STATUS" PROCESS. ; ; FUNCTION USES LOCAL VARUIABLE "CHTPI" AS THE BASIS FOR RETURNING ALL ; ; TRADING PARTNER INDICES. IF TRADING PARTNER IS INACTIVE, FUNCTION MOVES ; ; TO THE NEXT TRACKING GLOBAL TRADING PARTNER. ; ;..... ;  GETCHTPI N TRKI,CHTPI S CHTPI="" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .I (\$P(\$G(^CHMX277(TRKI,0)),"^",3)=1) S CHTPARR(CHTPI)="" Q  GETCLMI(FLAG) ; MATCH A DATE WITH THE ^CHMXCL(I) ENTRY ; DATE DATE FOR THE ^CHMXCL() SEARCH ; FLAG "&gt;","&lt;","=","&gt;" DATE SEARCH POSSIBILITIES N CHCNT,LSTI,EXIT,WDATE,RTN S RTN=0,EXIT=0 S LSTI=\$P(^CHMXCL(0),"^",3))+1 F CHCNT=1:1 Q:(LSTI="")!(EXIT) S LSTI=\$O(^CHMXCL(LSTI),-1) D .S WDATE=\$P(\$P(\$G(^CHMXCL(LSTI,0)),"^",1),".",1) ; FILE OPENED DATE .I ((FLAG="&lt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &gt; SPECIFIED DATE .E I ((FLAG="=")&amp;(WDATE=DATE)) S RTN=LSTI,EXIT=1 ; DATE = OR &gt; SPECIFIED DATE </pre>	

Routine Name	CHMXWB21
<pre> .E I ((FLAG="&gt;")&amp;(WDATE'&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; OR = SPECIFIED DATE Q RTN  ..... ; FILEMAN UPDATE "DR" STRING BUILD FUNCTION ; .....  SETDARR(STR)   N REG,DRDATA,TMP,IDX,DARR   S DARR="",DRDATA=""   F IDX=1:1 S TMP=\$P(STR,"^",IDX) Q:TMP="" D   .S REG=\$P(TMP,"=",1),VAL=\$P(TMP,"=",2)   .S DARR(REG)=VAL   S DRDATA=\$\$SETDR^CHHRLIBFM("DARR")   Q DRDATA  ..... ; RETRIEVE VALUES FROM THE TRACKING GLOBAL ; ; NEEDS: CHTPI: TRADING PARTNER INDEX TO USE ; ; RUNTYPE: STATUS TYPE BEING CREATED ; .....  GETTOFROM(CHTPI) ; RETRIEVE TO-FROM FILES FROM TRACKING GLOBAL   N JDX,NODE,TRKI   S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; CONVERT CHTPI FOR TRACKING GLOBAL ACCESS   S NODE=\$S(RUNTYPE="P":20,RUNTYPE="F":30) ; SELECT PENDING/FINAL STATUS NODE   S JDX="A",JDX=\$O(^CHMX277(TRKI,NODE,JDX),-1) ; LOOK AT LAST ENTRY FOR THE STATUS   ;W !,"GETTOFROM: CHTPI = ",CHTPI," NODE= ",NODE," JDX= ",JDX   Q:JDX=""   S CHFROM=(\$P(\$G(^CHMX277(TRKI,NODE,JDX,0)),"^",4)) ; NEW "FROM"= OLD "TO"   S:CHFROM'=0 CHFROM=CHFROM+1,CHTO=\$P(\$G(^CHMPAY(0)),"^",3) ; SET START/ END INDEX   ;W !,"GETTOFROM: FROM= ",CHFROM," TO= ",CHTO   Q  ..... ; SUPPORT FUNCTIONS FOR THE FILE GENERATION FUNCTIONS DEFINED ABOVE. ; .....  DATERANGE(FROM,TO) ;Prompt for a FROM and TO date range.   N MAXTO   D NOW^%DTC S MAXTO=\$E(%,1,7)   S (FROM,TO)="", (DONE,POP)=0   F Q:(DONE)!(POP) D   .S FROM=\$\$GETDATE^CHMXWBUT("ENTER START DATE: ") S:FROM=-1 POP=1 Q:POP   .I ((FROM&lt;1)!(FROM&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S TO=\$\$GETDATE^CHMXWBUT("ENTER END DATE: ") S:TO=-1 POP=1 Q:POP   .I ((TO&lt;FROM)!(TO&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S DONE=1   Q POP  GETSTVAL(CHCLEI) ; REVERSE POLISH METHOD FOR RETRIEVING CLAIM STATUS </pre>	

Routine Name	CHMXWB21
	<pre> N CHCLCI,CHCLBI,CHCLAI,CHCLI,TARGEI,STAT,PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S PDI=\$P(^CHMXCLE(CHCLEI,100),"^",2) S:PDI="" PDI=\$P(^CHMXCLE(CHCLEI,100),"^",4) I PDI="" Q 6 ;Assume 6 that it was rejected if no PDI found...data integrity issue S STAT=0,TARGEI=0 F Q:TARGEI=CHCLEI D .S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) .S CHIDXS=0,CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,"")) .S TARGEI=\$P(CHIDXS,"*",3) Q STAT  ..... ; EMAIL STATUS TO INTERESTED PARTIES ; .....  AUTOMM(TYPE) D NOW^%DTC S FMDATE=\$E(%,1,7) S CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL_ (Total Records = "_CHBRCCNT_") S CHNB=CHNB+1,ZML(CHNB)=" " I TYPE="EFILE" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: ***ERROR*** CREATING 277U STAT file!" E I TYPE="OK" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: Successful CREATION 277U STAT file!" S CHNB=CHNB+1,ZML(CHNB)="EDI BATCH NUMBER ^CHMXCL("_CHMXI_",0)" S XMTEXT="ZML(",XMSUB="277U WebMD..Success.."_FMDATE S XMDUZ=.5 S XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="" S XMY("FRANK.PELUSO@VA.GOV")="" D ^XMD Q  ..... ; THE FOLLOWING ROUTINES ARE PRESERVED FOR HISTORICAL PURPOSES ; ; THERE ARE COPIES OF THESE FUNCTIONS IN CHMXWB07,CHMXWB11,CHMXWB21 ; .....  ; EXFTP S CHFTPMSG=0 ;H 1000 ;DEF4399 ajm removed hang, not needed XCOM S X=\$ZF(-1,"SUBMIT DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP.COM /NAME=WEB_WS_FTP_JOB/NOPRINTER/USER=DSMMANAG/PARAM="_CHFILE) ;//////// UNREMARK/REMARK OUT CODE FOR PRODUCTION ////////// ;H 1200 ;DEF4399 ajm removed hang, not needed K ^CHMZHOLD("RKN_W277") Q ;DEF4399 ajm addd - files are copied, no longer ftp'd ; FTPCK S RD="DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP_JOB.LOG" ;LIVE C RD K CHFTPTIM,CHFTPWT S ZE="",QFLAG=0,CHFTPWT=\$P(^CHMDIC(741002.17,1,2),"^",10) F CHFTPTIM=1:1:CHFTPWT H 60 </pre>



Routine Name	CHMXWB21
<pre> O RD:"R" D Q:(QFLAG=1) ;RKN, 6-29-05, MOD READONLY .I '\$T C RD Q .F U RD R RDLIN D Q:(QFLAG=1)!( \$ZE["ENDOFIL"]) ..I (RDLIN["226 Transfer complete"]) S CHFTPMSG=1,QFLAG=1 Q ..I (RDLIN["Charged CPU time:"]) S QFLAG=1 Q FTPERR C RD S CHNB=2 S ZML(CHNB)="" S ZML(CHNB)=CHVMSFL S CHNB=CHNB+1 S ZML(CHNB)="" S CHNB=CHNB+1 S XMTEXT="ZML(" K CHFTPMSG,RD,RDLIN ; ; ; ; CALCULATE A DATE 90 DAYS IN THE PAST ; ; ; DOLDFLS D NOW^%DTC S X1=X,X2=-90 D C^%DTC D YX^%DTC Q ; ; ; Error Handler / personnel notification functions for the 277 status file ; ; generation process. ; ; ; VMSERR C IO D ^%ZTER ; ERRMM ; ;S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)="",CHNB=CHNB+1 ;S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 ;S ZML(CHNB)=" 277U WebMD file!" ;S XMDUZ=.5,XMY("10722")="" ;RKN LIVE ;S XMTEXT="ZML(",XMSUB="277U WebMD..Error.."_FMDATE S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)="",CHNB=CHNB+1 S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 S ZML(CHNB)=" CSTAT file!" ;S XMDUZ=.5,XMY("JAMES.NELSON5@VA.GOV")="" ;RKN LIVE ;S XMY("SREEDHAR.MADANAPALLE@MED.VA.GOV")="" S XMDUZ=.5,XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="",XMY("FRANK.PELUSO@VA.GOV")="" S XMTEXT="ZML(",XMSUB="277U..Error.."_DATESTAMP ; BAS -END MOD D ^XMD </pre>	

Routine Name	CHMXWB21
<p>Q</p> <p>.....  ; \$TEXT VARIABLE SETUP ROUTINES ;  .....</p> <p>GETHDRVAL  S LOADTYPE=\$S(RUNTYPE="H":"F",1:"I")  Q</p> <p>GETCLMVAL  N CHCLB0,CHCLB1,CHMPAY1,CHCLE60  N TMPDTE S TMPDTE=0  S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))  S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))  S CHCLE64=\$G(^CHMXCLE(CHCLEI,64))  S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))  S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID  S BPPN=""  ; 6.BILLING PROVIDER PAYER NUMBER  S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID  S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME  S BPFNAME="" ;</p> <p>9.BILLING PROVIDER FIRST NAME  S BPMNAME=""  ; 10.BILLING PROVIDER MIDDLE NAME  S BPNAMEX="" ;</p> <p>11.BILLING PROVIDER NAME SUFFIX  S SPFTID=""  ; 12.RENDERING PHYSICIAN FEDERAL TAX ID  S SPPN=""  ; 13.RENDERING PHYSICIAN PAYER NUMBER  S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID  S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME  S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME  S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME  S SPNAMEX="" ;</p> <p>18.RENDERING PHYSICIAN NAME SUFFIX  S EMPIDNUM=""  ; 19.EMPLOYER IDENTIFICATION NUMBER  S EMPNAME=""  ; 20.EMPLOYER NAME  S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))  S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID  S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME  S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME  S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME  S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX  S PATID=""  ; 26.PATIENT ID  S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME  S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME  S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME  S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX</p>	

Routine Name	CHMXWB21
	<p>S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH  S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER  S:"MF"[PATGENDR PATGENDR="" ; DO NOT  ALLOW PATIENT GENDER UNKNOWN  S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER  S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)'&gt;0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1))  ; 34.CHARGE AMT  S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT AMOUNT  S CAPD=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",4),1:"") ; 36.CLAIM ADJ/PAYMENT DATE  S CHKEFTDATE=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",10),1:"") ; 37.CHECK/EFT DATE  S CHKEFTNUM=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",16),1:"") ; 38.CHECK/EFT NUMBER  S BTYPE=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",7),1:"") ; 39.BILL TYPE  S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; 40.PAYER CLAIM ID NUMBER (PDI)  S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) ; PDI NOT POPULATED,  DERIVED CLAIM CTL NUM  S PATACT=\$P(\$G(^CHMXCLE(CHCLEI,0)),"^",2) ; 41.PATIENT ACCOUNT NUMBER  S PRSCNUM=""  ; 42.PHARMACY PRESCRIPTION NUMBER  S VOUCHID="" ;  43.VOUCHER IDENTIFIER  S LOCSYSID=""  ; 44.APP/LOCATION SYSTEM ID  S GRPNUM=""  ; 45.GROUP NUMBER  S CLMTYP=\$\$GETCLMTYP ; GET  I/P/D CLAIM TYPE  S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47.  PROFESSIONAL SERVICE START DATE  S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1  INSTITUTIONAL START/END  S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2  DENTAL START/END  S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2)  Q  GPSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL CLAIM START/END  DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S CHFI=0  S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI))  S STDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11),TMPDTE=STDATE ; TMPDTE IS LAST  START DATE  F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:'CHFI D ; CHECK ALL LI "FROM" DATES  .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11)  .I TDATE&lt;STDATE S STDATE=TDATE ; TRACK  "OLDEST" START DATE  .I TDATE&gt;TMPDTE S TMPDTE=TDATE ;  TRACK "MOST RECENT" START DATE  S EDATE=\$\$G SVCEND  S:(EDATE="")!(EDATE&lt;TMPDTE) EDATE=TMPDTE  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES  GISTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL CLAIM START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES</p>

Routine Name	CHMXWB21
S CHFI=0 S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GDSTARTEND() ; POPULATE EMDEON FIELD FOR DENTAL CLAIM START/END DATES N STDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GETSTCVAL(TYPE) S PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 4.PAYER CLAIM ID NUMBER(PDI) S:PCINUM="" PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S:TYPE="CLM" LICNUM="" ; 5.CLM LVL LINE ITEM CONTROL NUMBER S:TYPE="DTL" LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; SERVICE LINE NUMBER S DATERR="" ; 10.EMDEON SPECIFIC DATA IN ERROR S ESCODE="" ; 11.EMDEON SPECIFIC STATUS CODE Q  ..... ; 9/7/2011: modified Procedure Modifier #1 extract to field 4 from field 3; ..... GDTLVAR\$ ; DTL RECORD DATA GATHERING S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S CHCLF1=\$G(^CHMXCLF(CHCLFI,1)) S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID S:SQID="" SQID="NU" ; DEFAULT VALUE IF NOT PROVIDED S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT PROVIDED S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1 S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2 S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3 S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4 S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND LINE ITEM PROV. PAYMENT AMOUNT S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE	

Routine Name	CHMXWB21
<pre> S:RVNUCODE="9999" RVNUCODE="" ; DON'T OUTPUT 9999 VALUE S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,"^",8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE) S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 16.EMDEON CLAIM NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17. PROFESSIONAL SERVICE START/END DATE S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1 INSTITUTIONAL SERVICE START/END DATES S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL SERVICE START/END DATES S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2) Q  ..... ; LINE ITEM FUNCTIONS TO GATHER THE SPECIFIC DATA FIELDS ; .....  ..... ; EMDEON REQUIRES THE EARLIEST START DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE START DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN SEARCHES THROUGH LINE ITEMS FOR ANY EARLIER DATES. ; ; THE EARLIEST DATE FOUND IS RECORDED AS THE START DATE. ; .....  ..... ; DETERMINE THE CLAIM TYPE, I.E. PROFESSIONAL, INSTITUTIONAL, OR DENTAL ; 4010/5010 VERSION IDENTIFIERS ; 98 / 222 PROFESSIONAL CLAIMS ; 96 / 223 INSTITUTIONAL CLAIMS ; 97 / 224 DENTAL CLAIMS .....  GETCLMTYP() N TYPE ; VERSION-FUNCTIONAL TYPE NUMBER FROM ^CHMXCLA() S TYPE=\$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13) Q:(TYPE[222])!(TYPE[98]) "P" Q:(TYPE[223])!(TYPE[96]) "I" Q:(TYPE[224])!(TYPE[97]) "D" Q "U"  ..... ; PROFESSIONAL START/END DATES .....  GPDSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE S:EDATE="" EDATE=STDATE </pre>	

Routine Name	CHMXWB21
S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GIDSTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) S:STDATE="" STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) S:EDATE="" EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) S:EDATE="" EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GDDSTARTEND() ; EMDEON DENTAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE S:EDATE="" EDATE=STDATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  ..... ; EMDEON REQUIRES THE LAST END DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE END DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN 2)SEARCHES THROUGH LINE ITEMS FOR ANY LATER DATES. ; ; THE LATEST DATE FOUND IS RECORDED AS THE END DATE. ; ..... GSVCEND() ; POPULATE EMDEON REQUIRED FIELD FOR SERVICE END DATE N ENDATE,CHFI,TDATE S CHFI=0,CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S ENDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",12) ; ^CHMXCLF(I) END DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:'CHFI D ; CHECK ALL LI "TO" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",12) .I TDATE>ENDATE S ENDATE=TDATE ; TRACK "YOUNGEST" END DATE Q ENDATE  GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT Q 0  GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT Q 0  GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) Q 0  ..... ; DOCUMENTREC (UTILITY THAT HELPS IN DOCUMENTING THE RECORD GENERATION PROCESS) ; ; CREATES A FILE THAT CONTAINS THE RECORD INFORMATION FOR THE 5010 ;	

Routine Name	CHMXWB21
<pre> ; EMDEON STATUS RECORDS, INCLUDING THE HEADER, CLAIM, LINE ITEM, AND TRAILER ; ; RECORDS. THIS FUNCTION USES THE FIELD DESCRIPTORS TO DOCUMENT EACH FIELD IN ; ; THE RECORDS, I.E.: ; ; 1) RECORD NAME ; ; 2) STARTING LOCATION IN THE RECORD ; ; 3) LENGTH (WIDTH) OF THE FIELD ; ; 4) JUSTIFICATION WITHIN THE FIELD ; ; 5) THE VALUE (HARD CODED FIELDS) OR THE CACHE FILELOCATION FROM WHICH THE ; ; VALUE IS RETRIEVED. ; ..... DOCUMENTREC N DOCFILE,TMPIO,LN,HTABS,FTABS,RTYPE,RNAME,CHTYPE,STR,COLNMS S RTYPE="EMDEONHDR^EMDEONCLM^EMDEONSTC^EMDEONTRLR^EMDEONLI" ; NAMES OF \$TEXT TABLES S COLNMS="FIELD NAME^USE^DESC^PAD^JUST^VALUE" ; COLUMN HEADER NAMES S HTABS="35^39^45^50^57" ; HEADER TAB STOPS FOR THE FIELD DESCRIPTIONS S FTABS="36^39^46^52^57" ; FIELD TAB STOPS FOR DESCRIPTIONS S DOCPATH="CHAMPVA_EDl:[KERMIT.WEBMD]" S DOCFILE="DOC277_5010_"_DATESTAMP_.TXT" ; STATUS MAPPING DOCUMENTATION S DOCFILE=DOCPATH_DOCFILE ; OUTPUT THE \$TEXT TO A PRINTABLE FILE S FLAG=\$\$OFIL^CHMXWBUT(DOCFILE,"NWS") S TMPIO=\$IO U DOCFILE F CHTYPE=1:1 S RNAME=\$P(RTYPE,"^",CHTYPE) Q:RNAME="" D .W !!,?20,"HEALTH CARE CLEARING HOUSE """,RNAME,"" RECORD DEFINITIONS" ; TITLE .W !,\$P(COLNMS,"^",1),?\$P(HTABS,"^",1),\$P(COLNMS,"^",2),?\$P(HTABS,"^",2),\$P(COLNMS,"^",3),?\$P(HTABS,"^",3),?\$P(COLNMS,"^",4),?\$P(HTABS,"^",4),?\$P(COLNMS,"^",5),?\$P(HTABS,"^",5),?\$P(COLNMS,"^",6),?\$P(HTABS,"^",6),?\$P(COLNMS,"^",7) .F LN=1:1 S STR=\$T(@RNAME+LN) Q:STR["END OF RECORD" D ; READ \$TEXT DESCRIPTOR ..W !,\$P(STR,";",3),?\$P(FTABS,"^",1),\$P(STR,";",10),?\$P(FTABS,"^",2),\$P(STR,";",8),?\$P(FTABS,"^",3),?\$P(STR,";",7),?\$P(FTABS,"^",4),?\$P(STR,";",6),?\$P(FTABS,"^",5),?\$P(STR,";",4) U TMPIO D CLOSEFILE^CHMXWBUT(DOCFILE) ; CLOSE CURRENT FILE Q  ..... ; EMDEONHDR: Common Header for EMDEON STATUS Files ; ; A single header is generated for each output file. ; ; DESC: "FIELD NAME";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA TYPE"; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR(record # and text description) ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT, C= CENTER ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH (ANY PRINTABLE CHARACTER) ; ; NOTE: PAD CHAR="" IF NO CHARACTER IS BETWEEN THE SEMICOLONS (I.E. ;) ; </pre>	

Routine Name	CHMXWB21
	<pre> ; NO PADDING WILL OCCUR IF THIS IS SET UP THIS WAY ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; ; ; FORMATDATA TREATS THE PAD CHAR (;) AS A NULL, SO NO PADDING OCCURS. ; ; THIS WILL ALLOW USE OF THE FORMATDATA FUNCTION WITHOUT MODIFICATION BETWEEN ; ; PADDED AND NON-PADDED FIELDS. ; ..... ; 8/4/11 DLB "6. CREATION TIME" FROM \$(DATESTAMP,9,12) TO \$(DATESTAMP,9,14) ; ; 8/15/11 DLB "2. FILE GROUP ID" INSERTED THE DATESTAMP VALUE TO ENSURE UNIQUENESS ; ; 9/7/2011 DLB 12. LOAD TYPE CHANGED TO PROVIDE "F" WHEN HISTORICAL FILE GENERATED ; .....  EMDEONHDR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";DATA LENGTH/PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"HDR";3;L;;3AN;0;R; ;;2.FILE GROUP ID;\$\$(GROUPID'="":GROUPID,1:DATESTAMP);20;L;;20AN;4;R; ;;3.FILE GROUP SEQUENCE NUMBER;FILECNT;2;R;0;3N;24;R; ;;4.FILE GROUP COUNT;GRPCNT;2;R;0;3N;26;R; ;;5.CREATION DATE;\$(DATESTAMP,1,8);8;L;;8AN;28;R; ;;6.CREATION TIME;\$(DATESTAMP,9,14);6;L;;6N;36;R; ;;7.TRADING PARTNER ID;"VAFNH";10;L;;10AN;42;R; ;;8.SUBMITTER NAME;"VA, HEALTH ADMIN CENTER";30;L;;30AN;53;R; ;;9.PAYER CONTACT NAME;"";60;L;;60AN;83;O; ;;10.PAYER SUPPORT TELEPHONE NUMBER;"";10;L;;10N;143;O; ;;11.PAYER SUPPORT EMAIL ADDRESS;"";80;L;;80AN;153;O; ;;12.LOAD TYPE;LOADTYPE;1;L;;1AN;233;R; ;;13.PAYER UNIQUE FILE IDENTIFIER;DATESTAMP;20;L;;20AN;234;R; ;;14.FILE TYPE;"CStat";5;L;;5AN;254;R; ;;15.VERSION CODE;"03";2;L;;2AN;258;R; ;;16.RELEASE CODE;"00";2;L;;2AN;260;R; ;;18.END OF RECORD;  ..... ; EMDEON CLAIM IEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for E01 File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; This table is used as a block of data(a record) for each claim to be processed. ; ; Therefore, separating this from the header and trailer tables allows multiple ; ; calls to this table. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; </pre>



Routine Name	CHMXWB21
; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTPI) ; ; 8/5/11 DLB FIELDS 12,14,15: TYPO FIXED FOR PROVIDER INFO (ADDED ^ TO CHMXCLB) ; ; 8/5/11 DLB FIELD 41: CHANGED INDEX FROM ^CHMXCLE(I,100 TO ^CHMXCLE(I,0 ; ..... 	
EMDEONCLM ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA LENGTH/PATTERN";FIELD START LOCATION";FIELD USE ;;1.RECORD ID;"CLM";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.MAINTENANCE TYPE CODE;\$\$(\$G(RUNTYPE)="A":"021",\$G(RUNTYPE)="P":"021",\$G(RUNTYPE)="H":"030",1:"001");3;L;;3AN;19;R;  ;;5.BILLING PROVIDER FEDERAL TAX ID;BPFTI;9;L;;9N;22;C; ;;6.BILLING PROVIDER PAYER NUMBER;BPPN;50;L;;50AN;31;C; ;;7.BILLING PROVIDER NATIONAL ID;BPNID;10;L;;10N;81;C; ;;8.BILLING PROVIDER LAST NAME;BPLNAME;60;L;;60AN;91;C; ;;9.BILLING PROVIDER FIRST NAME;BPFNAME;35;L;;35AN;151;O; ;;10.BILLING PROVIDER MIDDLE NAME;BPMNAME;25;L;;25AN;186;O; ;;11.BILLING PROVIDER NAME SUFFIX;BPNAMEX;10;L;;10AN;211;O; ;;12.SERVICE PROVIDER FEDERAL TAX ID;SPFTID;9;L;;9N;221;C; ;;13.SERVICE PROVIDER PAYER NUMBER;SPPN;50;L;;50AN;230;C; ;;14.SERVICE PROVIDER NATIONAL ID;SPNID;10;L;;10N;280;C; ;;15.SERVICE PROVIDER LAST NAME;SPLNAME;60;L;;60AN;290;C; ;;16.SERVICE PROVIDER FIRST NAME;SPFNAME;35;L;;35AN;350;O; ;;17.SERVICE PROVIDER MIDDLE NAME;SPMNAME;25;L;;25AN;385;O; ;;18.SERVICE PROVIDER NAME SUFFIX;SPNAMEX;10;L;;10AN;410;O; ;;19.EMPLOYER IDENTIFICATION NUMBER;EMPIDNUM;80;L;;80AN;420;C; ;;20.EMPLOYER NAME;EMPNAME;60;L;;60AN;500;C; ;;21.SUBSCRIBER ID;SUBSCID;80;L;;80AN;580;C; ;;22.SUBSCRIBER LAST NAME;SUBLNAME;60;L;;60AN;660;C; ;;23.SUBSCRIBER FIRST NAME;SUBFNAME;35;L;;35AN;720;O; ;;24.SUBSCRIBER MIDDLE NAME;SUBMNAME;25;L;;25AN;755;O; ;;25.SUBSCRIBER NAME SUFFIX;SUBNAMEX;10;L;;10AN;780;O; ;;26.PATIENT ID;PATID;80;L;;80AN;790;O; ;;27.PATIENT LAST NAME;PATLNAME;60;L;;60AN;870;R; ;;28.PATIENT FIRST NAME;PATFNAME;35;L;;35AN;930;R; ;;29.PATIENT MIDDLE NAME;PATMNAME;25;L;;25AN;965;O; ;;30.PATIENT NAME SUFFIX;PATNAMEX;10;L;;10AN;990;O; ;;31.PATIENT DATE OF BIRTH;PATDOB;8;L;;8DT;1000;R; ;;32.PATIENT GENDER;PATGENDR;1;L;;1AN;1008;O; ;;33.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;1009;O; ;;34.CLAIM CHARGE AMOUNT;CLMCHRG;18;L;;18N;1059;R; ;;35.CLAIM PYMT AMT;CLMPMT;18;R;;18N;1077;C; ;;36.CLAIM ADJ/PAYMENT DATE;CAPD;8;L;;8DT;1095;C; ;;37.CHECK/EFT DATE;CHKEFTDATE;8;L;;8DT;1103;O; ;;38.CHECK/EFT NUMBER;CHKEFTNUM;16;L;;16AN;1111;O; ;;39.BILL TYPE;BTYPE;3;L;;3AN;1127;O; ;;40.PAYER CLAIM ID NUMBER;PCIDNUM;50;L;;50AN;1130;R; ;;41.PATIENT ACCOUNT NUMBER;PATACT;50;L;;50AN;1180;O; 	

Routine Name	CHMXWB21
	<pre> ;;42.PHARMACY PRESCRIPTION NUMBER;PRSCNUM;50;L;;50AN;1230;O; ;;43.VOUCHER IDENTIFIER;VOUCHID;50;L;;50AN;1280;O; ;;44.APP/LOCATION SYSTEM ID;LOCSYSID;50;L;;50AN;1330;O; ;;45.GROUP NUMBER;GRPNUM;50;L;;50AN;1380;O; ;;46.CLAIM SERVICE DATE START;CLMSTDT;8;L;;8DT;1430;R; ;;47.CLAIM SERVICE DATE END;CLMENDT;8;L;;8DT;1438;R; ;;48.END OF RECORD;  ..... ; EMDEON CLAIM STATUS RECORD DETAIL: Contains the Descriptions for all data to be ; ; gathered for Claim level and Line Level records. Used by \$TEXT to gather all ; ; data &amp; set up format. ; ; Each record is described in detail for the following Parameters: ; ; NOTE: ; ; For each CLM record, there must be at least one STC record. ; ; For a single CLM record, The following is true about DTL records: ; ; 1) Payer can send any number of DTL records (including NONE) ; ; 3) Each DTL record sent corresponds to Line Level info for the CLM record. ; ; 4) For each DTL record sent, there must be at least one STC record. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTPI) ; .....  EMDEONSTC;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"STC";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;3;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;13;R; ;;4.PAYER CLAIM ID NUMBER;PCINUM;50;L;;50AN;18;R; ;;5.LINE ITEM CONTROL NUMBER;LICNUM;50;L;;50AN;68;O; ;;6.STATUS INFORMATION EFF. DATE;DATESTAMP;14;L;;14N;118;R; ;;7.CLAIM STATUS CATEGORY CODE;RJCODE;3;L;;3AN;126;R; ;;8.CLAIM STATUS CODE;RJSTATUS;3;L;;3AN;129;R; ;;9.ENTITY CODE;ENTITY;3;L;;3AN;132;O; ;;10.DATA IN ERROR;DATERR;264;L;;264AN;135;O; ;;11.EMDEON STATUS CODE;ESCODE;5;L;;5AN;399;O; ;;13.END OF RECORD;  ..... ; EMDEONTRLR: Common Trailer for EMDEON Status Files ; ; NOTE: a single trailer record is generated for each output file. ; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; </pre>

Routine Name	CHMXWB21
	<pre> ; FIELD NAME: 277 File FIELD DESCRIPTOR ; ; LENGTH: 277 FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... EMDEONTRLR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"TRLR";4;L;;4AN;0;R; ;;2.RECORD COUNT;COUNT;10;R;;10N;4;R; ;;3.END OF RECORD;  ..... ; EMDEON LINE LEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for CRF File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; NOTE: there may be multiple line items for each claim record. This table ; ; defines the record to be generated for each line item. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; .....  EMDEONLI ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"DTL";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.PAYER CLAIM ID NUMBER;PCID;50;L;;50AN;19;R; ;;5.LINE ITEM CONTROL NUMBER;LICN;50;L;;50AN;69;R; ;;6.SERVICE QUALIFIER ID;SQID;2;L;;2AN;119;R; ;;7.SERVICE IDENTIFICATION CODE;SICOD;48;L;;48AN;121;R; ;;8.PROCEDURE MODIFIER 1;PRCM1;2;L;;2AN;131;O; ;;9.PROCEDURE MODIFIER 2;PRCM2;2;L;;2AN;133;O; ;;10.PROCEDURE MODIFIER 3;PRCM3;2;L;;2AN;135;O; ;;11.PROCEDURE MODIFIER 4;PRCM4;2;L;;2AN;137;O; ;;12.LI CHARGE AMOUNT;LICHGA;18;L;;18N;139;R; ;;13.LINE ITEM PROV. PAYMENT AMOUNT;LIPPA;18;L;;18N;157;R; ;;14.REVENUE CODE;RVNUCODE;48;L;;48AN;175;C; ;;15.QUANTITY(UNITS OF SERVICE);QTYUOS;15;L;;15N;223;O; ;;16.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;238;O; ;;17.SRVC START DATE;SVCSTDTE;8;L;;8N;288;R; ;;18.SRVC END DATE;SVCENDTE;8;L;;8N;306;R; </pre>

Routine Name	CHMXWB21
;;20.END OF RECORD;	
Modified Logic (Changes are in bold and highlighted)	
<pre> CHMXWB21 ;HAC/DLB;WEB 5010 277 UNSOLICITED STATUS/DRIVER PART 1;10/25/10 ;;1.0;CHAMPVA SYSTEM;;JULY 11,2011;Build 1 ;;V1.0; ;;CALLED BY ^CHMXG001 TO PERFORM THE "ACK" STATUS ;;CALLED BY TASKMAN TO PERFORM THE "PENDING" AND "FINAL" STATUS ;;CALLS ENTRY POINTS TO CHMXWB22,CHMXWBUT ;;HR-COB-Medicare-A/B-Begin-9372 (27-May-2010)  ** ** ** 10/18/2011 ADDED TSTBLDCLM() ENTRY POINT TO VALIDATE THE REJECT VALUES BEING ** REPORTED IN THE CSTAT FILE ** 10/20/2011 MODIFIED THE LINE() FUNCTION TO FIX INCORRECT REJECT STATUS ** REPORTING IN THE CSTAT FILE ** 10/21/2011 ADDED THE SIX DATE GATHERING FUNCTIONS TO SUPPORT THE RULE ** PROVIDED BY BUSINESS GROUP FOR I/P/D CLAIMS ** 10/21/2011 ADDED THE GETCLMTYP() FUNCTION TO DETERMINE THE I/P/D CLAIM TYPE ** FOR BOTH 4010 AND 5010 INCOMING CLAIMS ** 10/24/2011 CHANGED THE DATERANGE FUNCTION TO A EXISTING DATE FUNCTION ** IN THE UTILITY ROUTINE: CHMXWBUT.INT ** 10/25/2011 CHANGED THE RENDERING PHYSICIAN NPI RETRIEVAL TO GET THE DATA ** FROM THE CHMXCLE(I,64) TIN DATA FIELD. ** 10/25/2011 ADDED A TEST FOR THE "U" VALUE RECEIVED FROM EMDEON TO RETURN "" ** IF NOT A "M" OR "F" (ONLY ACCEPTED VALUES) ** 10/26/2011 REMOVED ERROR MESSAGE OUTPUT FOR ERROR TRAP HANDLER ** ** ** 10/26/2011 CHANGED THE INSTITUTIONAL AND DENTAL START/END DATES FOR CLAIM ** LEVEL REPORTING TO GET START FROM ^CHMXCLE(I,1),FIELD 1 AND END ** TO ^CHMXCLE(I,1) FIELD 2 ** 11/4/2011 CHANGED THE DATE CHECK ROUTINE GETCLMI() TO RETURN THE CORRECT ** DATE FOR THE DESIRED FLAG ** ; ;1.0 IS THE INITIAL VERSION (HIPAA Ready LLC) ;2.0 UPDATES TO THE STATUS REPORT PER HIPAA II SPECIFICATIONS. ; *** Filenames should not be hard coded in files. See ^CHMX277 TRACKING ; GLOBAL FILE for the destination directory/filenames for the 5010 status ; files. ; ; Q  ***** ; PRODUCTION ENTRY POINTS TO PROCESS THE CURRENT STATUS. ; *****  EPACK(CHMXI) ; SET UP THE FILE/BATCH INDEX (PROVIDED BY CHMXG001) ; CHMXI I INDEX INTO ^CHMXCL() N COUNT,DIRFILE,GRPCNT,FILECNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR,CNT </pre>	

Routine Name	CHMXWB21
	<pre> N MAXCNT,EICNT,CHA,CHMEDCOB,TPINT S FILECNT=0,RUNTYPE="A",GRPCNT=1,EICNT=0 S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER D INIT S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" S CHTPI=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:CHTPI="" ; GET CLAIM T/P FOR FILE BEING CHECKED S CHTPI=\$O(^CHMXTP("C",CHTPI,"")) Q:CHTPI="" D ; GET "LOCAL" TP ID S IDX=\$O(^CHMX277("B",CHTPI,0)) S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER I CHMEDCOB=0 S TPINT=\$P(^CHMX277(IDX,0),"^") I TPINT="" S CHMEDCOB=+\$P(\$G(^CHMXTP(TPINT,0)),"^",11) D GETCHCLEI(CHMXI,CHTPI) ; RETRIEVE ^CHMXCLE INDEX Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPPENDTST N TEST S TEST=1 D EPPEND Q  EPPEND N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR N CHNODEDEF,BREF,CHIJKVAL,DRDATA,TRKDATA,MAXCNT,CHMEDCOB,IDX ; FILEMAN UPDATE VARIABLES N CHTO,CHFROM,TRKI,EICNT S RUNTYPE="P",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER .S IDX=\$O(^CHMX277("B",CHTPI,0)) .S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .K CHTPARR .S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0) .S CHTPARR(CHTPI)="" ; SET CHTPARR UP FOR GETPENDEI .D GETTOFROM(CHTPI) Q:(CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPFINAL ; entry point to start Final Status Creation file N RUNTYPE,GRPCNT,EICNT,GROUPID,CHTPI,CHTPARR,CHFROM,CHTO,FILECNT </pre>

Routine Name	CHMXWB21
<pre> N TRKI N COUNT,DIRFILE,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,FMDATE N CHNODEDEF,BREF,CHJKVAL,DRDATA,TRKDATA,MAXCNT FILEMAN UPDATE VARIABLES S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .K CHTPARR S (CHFROM,CHTO,FILECNT,CHMEDCOB)=0 ; KILL TRADING PARTNER ARRAY; ASSUME NOT A "COB" TRADING PARTNER .S TRKI=\$O(^CHMX277("B",CHTPI,0)) ; GET TRACKING GLOBAL INDEX .I \$P(\$G(^CHMX277(TRKI,0)),"^",3)=0 Q .I \$P(^CHMX277(TRKI,0),"^",4)["COB" S CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .S CHTPARR(CHTPI)=" " .D GETTOFROM(CHTPI) Q:(^CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETFINALEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPREPTST N TEST S TEST=1 D EPREPLACEMENT Q  EPREPLACEMENT N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT,CHMEDCOB,IDX,CHTPX S RUNTYPE="H",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="REPLACEMENT" S TP=\$\$LISTTP^CHMXWB22 ; SELECT TRADING PARTNER S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER S IDX=\$O(^CHMX277("B",TP,0)) S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHXI="" .F S CHXI=\$O(^CHMXCL("B",IDATE,CHXI)) Q:CHXI="" D ..S CHMXI=CHXI ..I \$D(TEST) W !,"CHECKING:",\$\$DTCVRT(\$E(IDATE,1,7)) ; SHOW FILE DATES BEING PROCESSED ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" </pre>	

Routine Name	CHMXWB21
<pre> ..S CHTPX=\$P(\$G(^CHMXCLA(CHA,1)),",^",1) Q:CHTPX="" ; GET CLAIM T/P FOR FILE BEING CHECKED ..S CHTPX=\$O(^CHMXTP("C",CHTPX,"")) ; CHECK SELECTED VS CLAIM TP ID ..I TP=CHTPX D ; FILE MATCHES THE SELECTED TP ...S CHTPI=TP ; CHTPI IS USED DOWNSTREAM ...I \$D(TEST) W " PROCESSING:",\$\$DTCVRT(\$E(IDATE,1,7)) ...I \$D(TEST) W !," EPREPLACEMENT: CHMXI= ",CHMXI," CHA= ",CHA," CHTPI= ",CHTPI," TP= ",TP ...S CHTPARR(TP)="" D GETCHCLEI(CHMXI,CHTPI) ; IF TP VALID, PROCESS FILE I \$D(TEST) W !,"REPLACE: \$J= ",\$J," REC COUNT= ",\$G(^ZSC(\$J,0))," CHTPI= ",CHTPI," TP= ",TP Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  DTCVRT(DATE) ; CONVERT DATE TO EXTERNAL FORMAT N EXTDATE S EXTDATE=\$E(,DATE,4,5)_"-"_ \$E(,DATE,6,7)_"-"_(\$E(,DATE,1,3)+1700) Q EXTDATE  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPOVERTST N TEST S TEST=1 D EPOVERLAY Q  EPOVERLAY N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT S RUNTYPE="O",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="OVERLAY" S TP=\$\$LISTTP^CHMXWB22 D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHMXI="" .F S CHMXI=\$O(^CHMXCL("B",IDATE,CHMXI)) Q:CHMXI="" D ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S TPTIN=\$P(\$G(^CHMXCLA(CHA,1)),",^",1) Q:TPTIN="" ..I TP=\$O(^CHMXTP("C",TPTIN,"")) S CHTPARR(TP)="" D GETCHCLEI(CHMXI,TP) Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  ..... ; INIT PERFORMS LOCAL VARIABLE INITIALIZATION REQUIREMENTS FOR THE STATUS ; ; PROCESSES. ; </pre>	

Routine Name	CHMXWB21
	<pre> ..... INIT      ;           K ^ZSC(\$J) S ^ZSC(\$J,0)=0           K TMP           S GROUPID="",CHTPI=0           I '\$D(DATESTAMP) D                      ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS           .D NOW^%DTC           .S FMDATE=%           .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")           .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")           .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND           Q  .....           ; EPSTAT IS THE ENTRY POINT FOR THE ON-DEMAND REPORT FOR STATISICS ; .....  EPSTAT           D ONDEMAND^CHMXWB22           Q  EPREGEN           D REGEN^CHMXWB22           Q  .....           ; FILE CREATION FUNCTIONS ; .....  CREATEFILE ;           S MAXCNT=500000           ; HAPE POR 11/21/13 force max file size under 150 meg           I RUNTYPE="F" S MAXCNT=175000           D FILELOOP           Q FILELOOP      ; BUILD CLAIMS FROM ^ZSC(\$J) ARRAY (^CHMXCLE(I) VALUES)           N CHCLEI,PDI           D GETGRPCNT      ; GET ESTIMATED GROUP FILE COUNT           Q:GRPCNT=0           D GETFILE U DIRFILE ; CREATE ORIGINAL FILE           D BLDHDR          ; BLDHDR SETS COUNT=2 (SEE EMDEON SPEC)           S CHCLEI=0           F S CHCLEI=\$O(^ZSC(\$J,CHTPI,CHCLEI)) Q:CHCLEI="" D           .S EICNT=EICNT+1           .S STVAL=\$G(^ZSC(\$J,CHTPI,CHCLEI))           .D BLDCLM(CHCLEI,STVAL) ; GENERATE CLAIM/STC,DTL/STC RECORDS           .D CHKSEQ          ; CHECK FOR 500MB LIMIT, CREATE NEW FILE           D BLDTRL          ; BUILD THE TRAILER RECORD           D CLOSEFILE        ; CLOSE CURRENT FILE           Q </pre>



Routine Name	CHMXWB21
<pre> GETGRPCNT ; ESTIMATE THE TOTAL NUMBER OF FILES REQUIRED   N RECCNT,X   S RECCNT=0   S X=""   F S X=\$O(^ZSC(\$J,CHTPI,X)) Q:X="" S RECCNT=RECCNT+1   S GRPCNT=RECCNT\MAXCNT ; INTEGER DIVISION OF RECORD COUNT   S:RECCNT#MAXCNT&gt;0 GRPCNT=GRPCNT+1   ;W !,"GRPCNT=",GRPCNT   Q  CHKSEQ ; HANDLE THE FILE SEQUENCE NUMBERING REQUIREMENTS   I EICNT&gt;MAXCNT D     .D BLDTRL ; OUTPUT TRAILER RECORD     .D CLOSEFILE     .D GETFILE U DIRFILE     .D BLDHDR ; BLDHDR RESETS "COUNT" VARIABLE   Q  FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN   Q:"APF"[RUNTYPE ; A=ACK, P=PENDING, F=FINAL   N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA   S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^.03="_DIRFILE_".^04="_COUN T_".^05="_CHMXI   S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20,"TRKDATA=".02=1^.03="_DIRFILE_".^04="_CHTO _".^05="_CHFROM_".^06="_COUNT   S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30,"TRKDATA=".02=1^.03="_DIRFILE_".^04="_CHTO _".^05="_CHFROM_".^06="_COUNT   S BREF=FMDATE   S CHIKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE   S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT   D ADD^CHHRLIBFM(CHNODEDEF,CHIKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL   I RUNTYPE="F" N X,X1 F S X=\$O(^ZSC1(\$J,X)) Q:X="" F S X1=\$O(^ZSC1(\$J,X,X1)) Q:X1="" I \$D(^CHMPAY("E",X,X1)) S ^CHMPAY("E",X,X1)=1   Q  ..... ;TEST ENTRY POINT TO TEST THE BLDCLM FUNCTION FROM A PDI VALUE ; THERE ARE MULTIPLE CROSS REFERENCES (SOME DOCUMENTED, SOME UNDOCUMENTED ; THAT ARE UTILIZED IN THE CLAIM PROCESSING SYSTEM. THE CVTPDI AND TSTBLDCLM ; FUNCTIONS UTILIZE THE MAJORITY OF THESE CROSS REFERENCES. .....  CVTPDI(CHREFN)   N CHPCN,CHMXCLI,CHMXID,CHCLFI,SDATE,EDATE,CHCLFI,CHCLEI,IDX   I '\$D(^CHMXCLE("PDI",CHREFN)) W !,"NO ^CHMXCLE("PDI",,CHREFN,) XREF" Q   S CHPCN="" S1 S CHPCN=\$O(^CHMXCLE("PDI",CHREFN,CHPCN)) I CHPCN="" Q ;VALID PDI,NO DATA   S CHMXCLI=0 S2 S CHMXCLI=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI)) I 'CHMXCLI Q ;VALID PDI, NO DATA   S CHMXID="" S3 S CHMXID=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI,CHMXID)) I CHMXID="" Q </pre>	

Routine Name	CHMXWB21
<pre> S CHCLEI=\$P(CHMXID,"*",4) W !,"CLM SDATE [^CHMXCLE(I,1)),1]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) W !,"CLM EDATE [^CHMXCLE(I,1)),2]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) S CHCLFI=0 W !,"LINE ITEM DATE VALUES" F IDX=1:1 S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI D ; CHECK ALL LINE ITEM DATES .W !,IDX," ", " L/I SDATE [^CHMXCLF(I,1)),11]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) .W !," L/I END [^CHMXCLF(I,1)),12]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) .W !," THERAPY SDATE [^CHMXCLF(I,2.5)),1]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",1) .W !," LAST CERT DATE [^CHMXCLF(I,2.5)),2]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",2) Q  TSTBLDCLM(CHPDI) N CHPCN,CHCLI,CHCLAI,CHCLEI,STVAL,COUNT,PDI,BUFI,RUNTYPE,CLMTYPE,TEST S (STVAL,CHPCN)="",(COUNT,CHCLEI,CHCLAI,PDI)=0,TEST=1 S CHPCN=0,CHPCN=\$O(^CHMXCLE("PDI",CHPDI,CHPCN)) I CHPCN="" W !,"NO CHPCN VALUE" Q E W !,"PATIENT CONTROL # [^CHMXCLE("PDI",CHPDI,PCN)]= ",CHPCN ;  PATIENT CONTROL NUMBER S CHCLI=0,CHCLI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI)) I CHCLI="" W !,"NO CHCLI VALUE" Q S STVAL=0,STVAL=\$O(^CHMXCLE("A",CHCLI,STVAL)) W !,"XREF CLAIM STATUS= ",STVAL S BUFI="",BUFI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI)) I BUFI="" W !,"NO BUFFER INDEX VALUES" Q W !,"XREF [\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))] BUFFER INDICES= ",BUFI S CHCLEI=\$P(BUFI,"*",4) I CHCLEI="" W !,"NO CHCLEI VALUE RECOVERED" S RUNTYPE="H" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; RETRIEVE PDI S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) W !,"PDI VALUE FROM [^CHMXCLE(I,100)),2 OR 4]= ",PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),^",1) W !,"X12 837 BUFFER [CHMXCL(I)= ",?30,CHCLI W !,"TRANSACTION [CHMXCLA(I)]= ",?30,CHCLAI W !,"PROVIDER BUFFER [CHMXCLB(I)]= ",?30,CHCLBI W !,"PATIENT BUFFER [CHMXCLC(I)= ",?30,CHCLCI W !,"CLM BUFFER [CHMXCLE(I)= ",?30,CHCLEI W !,"L/I BUFFER [CHMXCLF(I)]= ",?30,\$O(^CHMXCLF("B",CHCLEI,0)) S CLMTYPE=\$\$GETCLMTYP W !,"VERSION-FUNCTIONAL TYPE [^CHMXCLA(I,0)),13]=\$ ", \$P(\$G(^CHMXCLA(CHCLAI,0)),^",13) W !,"CLAIM TYPE= ", \$\$S(CLMTYPE="I": "INSTITUTIONAL",CLMTYPE="P": "PROFESSIONAL",CLMTYPE="D": "DENTAL") ; I/P/D CLAIM W !,"STATUS (ACK/PENDING)=", \$\$S(\$D(^CHMPAY("C",PDI))=0: "ACK",1: "PENDING") D CVTPDI(CHPDI) W ! D BLDCLM(CHCLEI,"") Q </pre>	

Routine Name	CHMXWB21
<pre> ; BLDCLM: ENTRY POINT TO BUILD THE CSTAT RECORDS ; THIS FUNCTION PROVIDES AN EXTERNAL ENTRY POINT TO GENERATE THE ; EMDEON CLAIM, CLAIM STATUS, DETAIL, AND DETAIL STATUS RECORDS ..... </pre>	
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,'\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1),"^",4),CHKEFT=\$P(^1),"^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0),"^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") </pre>	

Routine Name	CHMXWB21
<pre> . D LINE Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^", 2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^", 4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D . S CHMPAY=0,OK=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)), "^", 2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)), "^", 2)=0 Q .. S OK=0 .. Q Q OK  ..... ; FUNCTIONS TO GATHER INFORMATION REGARDING REJECT REASONS ; .....  BLDACK ; POPULATE CHRJARR REJECT REASON/REJECT CODE VALUES K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY S REJCODES="" D GTRXRJRSN^CHMXWBUT(CHCLAI) ; TRANSACTION "101" NODE REJECT REASONS D GPRORJRSN^CHMXWBUT(CHCLBI) ; PROVIDER "101" NODE REJECT REASONS D GPATRJRSN^CHMXWBUT(CHCLCI) ; PATIENT "101" NODE REJECT REASONS D GCLMRJRSN^CHMXWBUT(CHCLEI) ; CLAIM "101" NODE REJECT REASONS I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" ;I \$D(TEST) W !,"BLDACK: RJCNT=",CHRJARR(0)," RJARR(1,1)=",CHRJARR(1,1) Q  ..... ; RECORD BUILDING FUNCTIONS: \$TEXT FILES PROVIDE FIELD DEFINITIONS ; .....  BLDHDR ;BUILD HEADER RECORD N LN,REC,STR,LOADTYPE S (STR,LN,REC)="",COUNT=2 ; EMDEON SPEC: REC. COUNT STARTS @ 2 D GETHDRVAL ; SET UP HEADER VARIABLES F LN=1:1 S STR=\$T(EMDEONHDR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="" Q  CLM ; N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID,</pre>	

Routine Name	CHMXWB21
<b>SUBNAME</b> N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN <b>AMEX,PATDOB</b> N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P <b>CIDNUM</b> N PATACCT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDY,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q	
LINE ; BUILD DTL RECORDS FOR LINE ITEMS N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE N CHJVAL,RJRSN,REC,STR,LN, <b>HACCLM</b> S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D .D GDTLVARS .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*"	
.I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),^") .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65*" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1*" Q ... S CHRJARR(1,1)="F0*1*"	
... Q .. Q .D BLDSTC("DTL") Q	
BLDSTC(TYPE) ; WRITE OUT THE STC RECORDS FROM CHRJARR REJECT ARRAY ; TYPE CLAIM OR DETAIL LEVEL EMDEON STATUS VALUES & LINE ITEM CONTROL NUMBER N PCINUM,LICNUM,DATERR,ESCODE N IDX,JDX,REJCODES,ECNT N LN,STR,REC S REJCODES=0 S ECNT=CHRJARR(0) ; FRONT END EDIT REJECT COUNT F IDX=1:1 Q:(IDX>ECNT) D ; PROVIDE STC RECORD FOR EACH ERROR	

Routine Name	CHMXWB21
<pre> .F JDX=1:1 Q:\$G(CHRJARR(IDX,JDX))="" D ; GET EACH REJECT VALUE STORED ..S (RJCODE,RJSTATUS,ENTITY)="" ..D GETSTCVAL(TYPE) ; EMDEON STATUS RECORD VALUES FOR CLM OR DTL ..S REJCODES=\$G(CHRJARR(IDX,JDX)) ..;W:\$D(TEST) !,"BLDSTC: REJCODE STR=",REJCODES,! ; DEBUG: ONLY OUTPUTS IF "TEST" IS DEFINED ..S RJCODE=\$P(REJCODES,"*",1),RJSTATUS=\$P(REJCODES,"*",2),ENTITY=\$P(REJCODES, "*,3) ..S REC="" ..F LN=1:1 S STR=\$T(EMDEONSTC+LN) Q:STR["END OF RECORD" D ; GENERATE THE STATUS (STC) RECORD ...I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ...E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 Q  BLDTRL ; TRAILER RECORD N LN,REC,STR S COUNT=COUNT-2 ; ADJUST COUNT FOR START OF 2 (SEE EMDEON SPEC) S (STR,LN,REC)="" F LN=1:1 S STR=\$T(EMDEONTRLR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ; 10/20/2010 ADDED "ELSE" W REC,! S REC="" Q  ..... ; GENERIC FUNCTIONS SUPPORTING THE RECORD BUILDING ; .....  GETFILE() N DIRPATH S EICNT=0,FILECNT=FILECNT+1 S DIRPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP D NOW^%DTC S FMDATE=% S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT" E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT" ; CHMEDCOB SET IN CHMXWT1 E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME I RUNTYPE="H" S DIRFILE=DIRFILE_"_H" I RUNTYPE="O" S DIRFILE=DIRFILE_"_O" S FLAG=\$\$OFILE^CHMXWBUT(DIRFILE,"NWS") I FLAG D VMSERR Q "" S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q  TST </pre>	

Routine Name	CHMXWB21
<pre> N FMDATE,TIME S FMDATE=3110901.1234 W !,FMDATE S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") W !,FMDATE S TIME=\$E(FMDATE,9,14) W !,TIME S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") W !,TIME Q  CLOSEFILE C DIRFILE Q  GETCHCLEI(CHCLI,CTPI) ; BUILD ARRAY OF CLAIM INDICES (CHMXCLE(I) VALUES) ; CHCLI I INDEX FOR ^CHMXCL() FILE (CLAIM FILE RECEIVED) N STAT,CHCLAI,PDI,CHIDX,CHCLEI,RECCNT,CTPI S CTPI=0 S STAT="",STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) I \$D(TEST) W:STAT="" !,"NULL STAT" Q:STAT="" ; RETRIEVE STATUS VALUE S CHCLAI="",CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) I \$D(TEST) W:CHCLAI="" !,"NULL CHCLAI" Q:CHCLAI="" ; RETRIEVE THE ^CHMXCLA(I) VALUE I \$D(TEST) W !," GETCHCLEI: READING:",CHCLI," STATUS= ",STAT," ^CHMXCLA(I)= ",CHCLAI," TP= ",CTPI S STAT="" F S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) Q:STAT="" D .S CHCLAI="" .F S CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) Q:CHCLAI="" D ..S PDI=0 ..F S PDI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI)) Q:'PDI D ...S CHIDX=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,0)) ...Q:CHIDX="" ; PDI  RETRIEVES BUFFER IDXs ...S CHCLEI=\$P(CHIDX,"*",3) ...S ^ZSC(\$J,CTPI,CHCLEI)=STAT,RECCNT=\$I(^ZSC(\$J,0)) Q  GETTP(CHA) N TPID,ID S ID="" S TPID=\$P(\$G(^CHMXCLA(CHA,1)), "^",1) S:TPID]"" ID=\$O(^CHMXTP("C",TPID,"")) Q IDER  GETPENDEI(CHFROM,CHTO) ; GET CHCLEI FROM PENDING ^CHMPAY(I) FROM/TO VALUES K ^ZSC(\$J) N IDX,CHJK,PDI,CHCLEI,CHCLAI,CTPI,RECCNT S RECCNT=0 F IDX=CHFROM:1 Q:IDX&gt;CHTO D .S CHJK=\$O(^CHMPAY(IDX,"PDI",0)) Q:CHJK="" .S PDI=\$P(^CHMPAY(IDX,"PDI",CHJK,0), "^",1) Q:PDI="" .S CHCLEI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",6) Q:CHCLEI="" .S CHCLAI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",3) Q:CHCLAI="" .S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM .I CHCLEI]"" I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1 S ^ZSC(\$J,0)=RECCNT </pre>	

Routine Name	CHMXWB21
<pre> Q  GETFINALEI(CHFROM,CHTO) ; GET CHCLEI FOR FINAL CLAIM STATUS K ^ZSC(\$J),^ZSC1(\$J) N CHCLEI,CHCLAI,CTPI,RECCNT,CLC,CLB,CHPIDX,CHPDATE,CHPPDI S RECCNT=0 S %H=\$H-180 D YMD^%DTC S CHPDATE=X F S CHPDATE=\$O(^CHMPAY("E",CHPDATE)) Q:CHPDATE="" S CHPIDX="" F S CHPIDX=\$O(^CHMPAY("E",CHPDATE,CHPIDX)) Q:CHPIDX="" I \$G(^CHPIDX)'=1 D . S CHPPDI=\$O(^CHMPAY(CHPIDX,"PDI",0)) Q:CHPPDI="" . I CHPPDI=\$P(\$G(^CHMPAY(CHPIDX,"PDI",PDI,0)),^") . I CHPPDI="" Q . S CHCLEI=\$O(^CHMXCLE("D",CHPPDI,"")) I CHCLEI="" Q . I \$\$FINAL(CHCLEI) D .. S CLC=\$P(^CHMXCLE(CLE,0),^") .. S CLB=\$P(^CHMXCLC(CLC,0),^") .. S CHCLAI=\$P(^CHMXCLB(CLB,0),^") .. S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM .. I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1,^ZSC1(\$J,CHPDATE,CHPIDX)="" S ^ZSC(\$J,0)=RECCNT Q  ; ;..... ; GETCHTPI RETURN THE NEXT TRADING PARTNER INDEX BASED ON THE LAST INDEX ; ; USED. THIS FUNCTRION IS CALLED BY THE "PENDING STATUS" PROCESS. ; ; FUNCTION USES LOCAL VARUIABLE "CHTPI" AS THE BASIS FOR RETURNING ALL ; ; TRADING PARTNER INDICES. IF TRADING PARTNER IS INACTIVE, FUNCTION MOVES ; ; TO THE NEXT TRACKING GLOBAL TRADING PARTNER. ; ;.....  GETCHTPI N TRKI,CHTPI S CHTPI="" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .I (\$P(\$G(^CHMX277(TRKI,0)),^",3)=1) S CHTPARR(CHTPI)="" Q  GETCLMI(,DATE,FLAG) ; MATCH A DATE WITH THE ^CHMXCL(I) ENTRY ; DATE DATE FOR THE ^CHMXCL() SEARCH ; FLAG "&gt;","&lt;","&lt;","OR "&gt;" DATE SEARCH POSSIBILITIES N CHCNT,LSTI,EXIT,WDATE,RTN S RTN=0,EXIT=0 S LSTI=(\$P(^CHMXCL(0),^",3))+1 F CHCNT=1:1 Q:(LSTI="")!(EXIT) S LSTI=\$O(^CHMXCL(LSTI),-1) D .S WDATE=\$P(\$P(\$G(^CHMXCL(LSTI,0)),^",1),",1) ; FILE OPENED DATE .I ((FLAG="&lt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; SPECIFIED DATE .I ((FLAG="&gt;")&amp;(WDATE&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &gt; SPECIFIED DATE .I ((FLAG="&lt;")&amp;(WDATE'=DATE)) S RTN=LSTI,EXIT=1 ; DATE = OR &gt; SPECIFIED DATE .I ((FLAG="&gt;")&amp;(WDATE'=DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; OR = SPECIFIED DATE Q RTN  ; ;..... ; FILEMAN UPDATE "DR" STRING BUILD FUNCTION ; </pre>	



Routine Name	CHMXWB21
	<pre> ***** ***** SETDARR(STR)   N REG,DRDATA,TMP,IDX,DARR   S DARR="",DRDATA=""   F IDX=1:1 S TMP=\$P(STR,"^",IDX) Q:TMP="" D   .S REG=\$P(TMP,"=",1),VAL=\$P(TMP,"=",2)   .S DARR(REG)=VAL   S DRDATA=\$\$SETDR^CHHRLIBFM("DARR")   Q DRDATA  ***** ***** ; RETRIEVE VALUES FROM THE TRACKING GLOBAL ; ; NEEDS: CHTPI: TRADING PARTNER INDEX TO USE ; ; RUNTYPE: STATUS TYPE BEING CREATED ; ***** ***** GETTOFROM(CHTPI) ; RETRIEVE TO-FROM FILES FROM TRACKING GLOBAL   N JDX,NODE,TRKI   S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; CONVERT CHTPI FOR TRACKING GLOBAL ACCESS   S NODE=\$S(RUNTYPE="P":20,RUNTYPE="F":30) ; SELECT PENDING/FINAL STATUS NODE   S JDX="A",JDX=\$O(^CHMX277(TRKI,NODE,JDX),-1) ; LOOK AT LAST ENTRY FOR THE STATUS   ;W !,"GETTOFROM: CHTPI = ",CHTPI," NODE= ",NODE," JDX= ",JDX   Q:JDX=""   S CHFROM=(\$P(\$G(^CHMX277(TRKI,NODE,JDX,0)),"^",4)) ; NEW "FROM"= OLD "TO"   S:CHFROM=0 CHFROM=CHFROM+1,CHTO=\$P(\$G(^CHMPAY(0)),"^",3) ; SET START/ END INDEX   ;W !,"GETTOFROM: FROM= ",CHFROM," TO= ",CHTO   Q  ***** ***** ; SUPPORT FUNCTIONS FOR THE FILE GENERATION FUNCTIONS DEFINED ABOVE. ; ***** ***** DATERANGE(FROM,TO) ;Prompt for a FROM and TO date range.   N MAXTO   D NOW^%DTC S MAXTO=\$E(%,1,7)   S (FROM,TO)="", (DONE,POP)=0   F Q: (DONE)!(POP) D   .S FROM=\$\$GETDATE^CHMXWBUT("ENTER START DATE: ") S:FROM=-1 POP=1 Q:POP   .I ((FROM&lt;1)!(FROM&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S TO=\$\$GETDATE^CHMXWBUT("ENTER END DATE: ") S:TO=-1 POP=1 Q:POP   .I ((TO&lt;FROM)!(TO&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S DONE=1   Q POP  GETSTVAL(CHCLEI) ; REVERSE POLISH METHOD FOR RETRIEVING CLAIM STATUS   N CHCLCI,CHCLBI,CHCLAI,CHCLI,TARGEI,STAT,PDI   S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES   S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1)   S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1)   S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) </pre>

Routine Name	CHMXWB21
<pre> S PDI=\$P(^CHMXCLE(CHCLEI,100),"^",2) S:PDI="" PDI=\$P(^CHMXCLE(CHCLEI,100),"^",4) I PDI="" Q 6 ;Assume 6 that it was rejected if no PDI found...data integrity issue S STAT=0,TARGEI=0 F Q:TARGEI=CHCLEI D .S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) .S CHIDXS=0,CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,"")) .S TARGEI=\$P(CHIDXS,"*",3) Q STAT  ..... ; EMAIL STATUS TO INTERESTED PARTIES ; .....  AUTOMM(TYPE) D NOW^%DTC S FMDATE=\$E(%,1,7) S CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL_ (Total Records = "_CHBRCCNT_") S CHNB=CHNB+1,ZML(CHNB)="" I TYPE="EFILE" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: ***ERROR*** CREATING 277U STAT file!" E I TYPE="OK" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: Successful CREATION 277U STAT file!" S CHNB=CHNB+1,ZML(CHNB)="EDI BATCH NUMBER ^CHMXCL("_CHMXI_",0)" S XMTEXT="ZML(",XMSUB="277U WebMD..Success.."_FMDATE S XMDUZ=.5 S XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="" S XMY("FRANK.PELUSO@VA.GOV")="" D ^XMD Q  ..... ; THE FOLLOWING ROUTINES ARE PRESERVED FOR HISTORICAL PURPOSES ; ; THERE ARE COPIES OF THESE FUNCTIONS IN CHMXWB07,CHMXWB11,CHMXWB21 ; .....  ; EXFTP S CHFTPMSG=0 ;H 1000 ;DEF4399 ajm removed hang, not needed XCOM S X=\$ZF(-1,"SUBMIT DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP.COM /NAME=WEB_WS_FTP_JOB/NOPRINTER/USER=DSMMANAG/PARAM="_CHFILE) ;//////// UNREMARK/REMARK OUT CODE FOR PRODUCTION ////////// ;H 1200 ;DEF4399 ajm removed hang, not needed K ^CHMZHOLD("RKN_W277") Q ;DEF4399 ajm addd - files are copied, no longer ftp'd ; FTPCK S RD="DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP_JOB.LOG" ;LIVE C RD K CHFTPTIM,CHFTPWT S ZE="",QFLAG=0,CHFTPWT=\$P(^CHMDIC(741002.17,1,2),"^",10) F CHFTPTIM=1:1:CHFTPWT H 60 O RD:"R" D Q:(QFLAG=1) ;RKN, 6-29-05, MOD READONLY .I '\$T C RD Q .F U RD R RDLIN D Q:(QFLAG=1)!( \$ZE["ENDOFIL") ..I (RDLIN["226 Transfer complete") S CHFTPMSG=1,QFLAG=1 Q ..I (RDLIN["Charged CPU time:") S QFLAG=1 Q </pre>	

Routine Name	CHMXWB21
<pre> FTPERR C RD   S CHNB=2   S ZML(CHNB)=" "   S ZML(CHNB)=CHVMSFL   S CHNB=CHNB+1   S ZML(CHNB)=" "   S CHNB=CHNB+1   S XMTEXT="ZML("   K CHFTPMSG,RD,RDLINE   ;   .....   ; CALCULATE A DATE 90 DAYS IN THE PAST ;   .....  DOLDFLS D NOW^%DTC   S X1=X,X2=-90   D C^%DTC   D YX^%DTC   Q   ;    .....   ; Error Handler / personnel notification functions for the 277 status file ;   ; generation process. ;   .....  VMSERR C IO       D ^%ZTER   ; ERRMM ;   ;S CHERRMSG=1,CHNB=2,ZML(CHNB)=" ",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)=" ",CHNB=CHNB+1   ;S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1   ;S ZML(CHNB)=" 277U WebMD file!"   ;S XMDUZ=.5,XMY("10722")=" " ;RKN LIVE   ;S XMTEXT="ZML(",XMSUB="277U WebMD..Error.."_FMDATE   S CHERRMSG=1,CHNB=2,ZML(CHNB)=" ",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)=" ",CHNB=CHNB+1   S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1   S ZML(CHNB)=" CSTAT file!"   ;S XMDUZ=.5,XMY("JAMES.NELSON5@VA.GOV")=" " ;RKN LIVE   ;S XMY("SREEDHAR.MADANAPALLE@MED.VA.GOV")=" "   S XMDUZ=.5,XMY("274577")=" ",XMY("246183")=" "   S XMY("AHMED.MAGDUB@VA.GOV")=" ",XMY("FRANK.PELUSO@VA.GOV")=" "   S XMTEXT="ZML(",XMSUB="277U..Error.."_DATESTAMP   ; BAS -END MOD   D ^XMD   Q    .....   ; \$TEXT VARIABLE SETUP ROUTINES ;   ..... </pre>	

Routine Name	CHMXWB21
<p>GETHDRVAL  S LOADTYPE=\$S(RUNTYPE="H": "F", 1: "I")  S (PAYNAME, PAYPHONE)=""  <b>I RUNTYPE="F" S PAYNAME="CHAMPVA CUSTOMER SERVICE", PAYPHONE="8007338387"</b>  Q</p> <p>GETCLMVAL  N CHCLB0, CHCLB1, CHMPAY1, CHCLE60  N TMPDTE S TMPDTE=0  S CHCLB0=\$G(^CHMXCLB(CHCLBI, 0))  S CHCLB1=\$G(^CHMXCLB(CHCLBI, 1))  S CHCLE64=\$G(^CHMXCLE(CHCLEI, 64))  <b>S: RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI, 1))</b>  S BPFTI=\$P(CHCLB0, "^", 2) ; 5.BILLING PROVIDER FEDERAL TAX ID  S BPPN=""  ; 6.BILLING PROVIDER PAYER NUMBER  S BPNID=\$P(CHCLB0, "^", 13) ; 7.BILLING PROVIDER NATIONAL ID  S BPLNAME=\$P(CHCLB0, "^", 3) ; 8.BILLING PROVIDER LAST NAME  S BPFNAME="" ;  9.BILLING PROVIDER FIRST NAME  S BPMNAME=""  ; 10.BILLING PROVIDER MIDDLE NAME  S BPNAMEX="" ;  11.BILLING PROVIDER NAME SUFFIX  S SPFTID=""  ; 12.RENDERING PHYSICIAN FEDERAL TAX ID  S SPPN=""  ; 13.RENDERING PHYSICIAN PAYER NUMBER  S SPNID=\$P(CHCLE64, "^", 4) ; 14.RENDERING PHYSICIAN NATIONAL ID  S SPLNAME=\$P(CHCLE64, "^", 1) ; 15.RENDERING PHYSICIAN NAME  S SPFNAME=\$P(CHCLE64, "^", 2) ; 16.RENDERING PHYSICIAN FNAME  S SPMNAME=\$P(CHCLE64, "^", 3) ; 17.RENDERING PHYSICIAN MNAME  S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX  S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER  S EMPNAME="" ; 20.EMPLOYER NAME  S CHCLC0=\$G(^CHMXCLC(CHCLCI, 0))  S SUBSCID=\$P(CHCLC0, "^", 4) ; 21.SUBSCRIBER ID  S SUBLNAME=\$P(CHCLC0, "^", 5) ; 22.SUBSCRIBER LAST NAME  S SUBFNAME=\$P(CHCLC0, "^", 6) ; 23.SUBSCRIBER FIRST NAME  S SUBMNAME=\$P(CHCLC0, "^", 7) ; 24.SUBSCRIBER MIDDLE NAME  S SUBNAMEX=\$P(CHCLC0, "^", 8) ; 25.SUBSCRIBER NAME SUFFIX  S PATID="" ; 26.PATIENT ID  S PATLNAME=\$P(CHCLC0, "^", 5) ; 27.PATIENT LAST NAME  S PATFNAME=\$P(CHCLC0, "^", 6) ; 28.PATIENT FIRST NAME  S PATMNAME=\$P(CHCLC0, "^", 7) ; 29.PATIENT MIDDLE NAME  S PATNAMEX=\$P(CHCLC0, "^", 8) ; 30.PATIENT NAME SUFFIX  S PATDOB=\$P(CHCLC0, "^", 9) ; 31.PATIENT DATE OF BIRTH  S PATGENDR=\$P(CHCLC0, "^", 10) ; 32.PATIENT GENDER  S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN  S ECLMNUM=\$P(^CHMXCLE(CHCLEI, 3), "^", 5) ; 33.EMDEON CLAIM NUMBER  S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI, 2), "^", 1)'&gt;0:"0.00", 1:\$P(^CHMXCLE(CHCLEI, 2), "^", 1))  ; 34.CHARGE AMT  <b>S CLMPMT=\$S(RUNTYPE="F": \$P(CHMPAY1, "^", 1), 1:"0.00") ; 35.CLAIM PAYMENT</b></p>	

Routine Name	CHMXWB21
<b>AMOUNT</b> <b>S CAPD=\$\$(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b> <b>S CHKEFTDATE=\$\$(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$\$(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$\$(RUNTYPE="F":",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER S VOUCHID="" ; 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47. PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1 INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2 DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q  GPSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S STDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",11),TMPDTE=STDATE ; TMPDTE IS LAST START DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:'CHFI D ; CHECK ALL LI "FROM" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",11) .I TDATE<STDATE S STDATE=TDATE ; TRACK "OLDEST" START DATE .I TDATE>TMPDTE S TMPDTE=TDATE ; TRACK "MOST RECENT" START DATE S EDATE=\$\$GSVCEND S:(EDATE="")!(EDATE<TMPDTE) EDATE=TMPDTE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES  GISTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES  GDSTARTEND() ; POPULATE EMDEON FIELD FOR DENTAL CLAIM START/END DATES	

Routine Name	CHMXWB21
N STDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GETSTCVAL(TYPE) S PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 4.PAYER CLAIM ID NUMBER(PDI) S:PCINUM="" PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S DATERR="" ; 10.EMDEON SPECIFIC DATA IN ERROR S ESCODE="" ; 11.EMDEON SPECIFIC STATUS CODE <b>I RUNTYPE="F" D Q</b> <b>. I \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICNUM=\$P(^",^",23)</b> <b>. I \$G(LICNUM)="" S LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2)</b> S:TYPE="CLM" LICNUM="" ; 5.CLM LVL LINE ITEM CONTROL NUMBER S:TYPE="DTL" LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; SERVICE LINE NUMBER Q  ..... ; 9/7/2011: modified Procedure Modifier #1 extract to field 4 from field 3; .....  GDTLVAR\$ ; DTL RECORD DATA GATHERING S LICN="" S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S CHCLF1=\$G(^CHMXCLF(CHCLFI,1)) I RUNTYPE="F",\$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICN=\$P(^",^",23) I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID S:SQID="" SQID="NU" ; DEFAULT VALUE IF NOT PROVIDED S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT PROVIDED S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1 S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2 S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3 S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4 S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND LINE ITEM PROV. PAYMENT AMOUNT S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE S:RVNUCODE="9999" RVNUCODE="" ; DON'T OUTPUT 9999 VALUE S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE) S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE)	

Routine Name	CHMXWB21
<pre> S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 16.EMDEON CLAIM NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17. PROFESSIONAL SERVICE START/END DATE S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1 INSTITUTIONAL SERVICE START/END DATES S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL SERVICE START/END DATES S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2) Q  ..... ; LINE ITEM FUNCTIONS TO GATHER THE SPECIFIC DATA FIELDS ; .....  ..... ; EMDEON REQUIRES THE EARLIEST START DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE START DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN SEARCHES THROUGH LINE ITEMS FOR ANY EARLIER DATES. ; ; THE EARLIEST DATE FOUND IS RECORDED AS THE START DATE. ; .....  ..... ; DETERMINE THE CLAIM TYPE, I.E. PROFESSIONAL, INSTITUTIONAL, OR DENTAL ; 4010/5010 VERSION IDENTIFIERS ; 98 / 222 PROFESSIONAL CLAIMS ; 96 / 223 INSTITUTIONAL CLAIMS ; 97 / 224 DENTAL CLAIMS .....  GETCLMTYP() N TYPE ; VERSION-FUNCTIONAL TYPE NUMBER FROM ^CHMXCLA() S TYPE=\$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13) Q:(TYPE[222]!(TYPE[98] "P" Q:(TYPE[223]!(TYPE[96] "I" Q:(TYPE[224]!(TYPE[97] "D" Q "U"  ..... ; PROFESSIONAL START/END DATES .....  GPDSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE S:EDATE="" EDATE=STDATE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES  GIDSTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) </pre>	

Routine Name	CHMXWB21
S:STDATE="" STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) S:EDATE="" EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) S:EDATE="" EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  GDDSTARTEND() ; EMDEON DENTAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE S:EDATE="" EDATE=STDATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  ..... ; EMDEON REQUIRES THE LAST END DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE END DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN 2)SEARCHES THROUGH LINE ITEMS FOR ANY LATER DATES. ; ; THE LATEST DATE FOUND IS RECORDED AS THE END DATE. ; .....  GSVCEND() ; POPULATE EMDEON REQUIRED FIELD FOR SERVICE END DATE N ENDATE,CHFI,TDATE S CHFI=0,CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S ENDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",12) ; ^CHMXCLF(I) END DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:'CHFI D ; CHECK ALL LI "TO" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",12) .I TDATE>ENDATE S ENDATE=TDATE ; TRACK "YOUNGEST" END DATE Q ENDATE  GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT Q 0  GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT Q 0  GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) Q 0  ..... ; DOCUMENTREC (UTILITY THAT HELPS IN DOCUMENTING THE RECORD GENERATION PROCESS) ; ; CREATES A FILE THAT CONTAINS THE RECORD INFORMATION FOR THE 5010 ; ; EMDEON STATUS RECORDS, INCLUDING THE HEADER, CLAIM, LINE ITEM, AND TRAILER ; ; RECORDS. THIS FUNCTION USES THE FIELD DESCRIPTORS TO DOCUMENT EACH FIELD IN ; ; THE RECORDS, I.E.: ; ; 1) RECORD NAME ; ; 2) STARTING LOCATION IN THE RECORD ; ; 3) LENGTH (WIDTH) OF THE FIELD ; ; 4) JUSTIFICATION WITHIN THE FIELD ; ; 5) THE VALUE (HARD CODED FIELDS) OR THE CACHE FILELOCATION FROM WHICH THE	



Routine Name	CHMXWB21
<pre> ; ; VALUE IS RETRIEVED. ; ..... DOCUMENTREC N DOCFILE,TMPIO,LN,HTABS,FTABS,RTYPE,RNAME,CHTYPE,STR,COLNMS S RTYPE="EMDEONHDR^EMDEONCLM^EMDEONSTC^EMDEONTRLR^EMDEONLI" ; NAMES OF \$TEXT TABLES S COLNMS="FIELD NAME^USE^DESC^PAD^JUST^VALUE" ; COLUMN HEADER NAMES S HTABS="35^39^45^50^57" ; HEADER TAB STOPS FOR THE FIELD DESCRIPTIONS S FTABS="36^39^46^52^57" ; FIELD TAB STOPS FOR DESCRIPTIONS S DOCPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" S DOCFILE="DOC277_5010_"_DATESTAMP_.TXT" ; STATUS MAPPING DOCUMENTATION S DOCFILE=DOCPATH_DOCFILE ; OUTPUT THE \$TEXT TO A PRINTABLE FILE S FLAG=\$\$OFIL^CHMXWBUT(DOCFILE,"NWS") S TMPIO=\$IO U DOCFILE F CHTYPE=1:1 S RNAME=\$P(RTYPE,"^",CHTYPE) Q:RNAME="" D .W !,?20,"HEALTH CARE CLEARING HOUSE """,RNAME,"" RECORD DEFINITIONS" ; TITLE .W !,\$P(COLNMS,"^",1),?\$P(HTABS,"^",1),\$P(COLNMS,"^",2),?\$P(HTABS,"^",2),\$P(COLNMS,"^",3),?\$P(HTABS,"^",3),?\$P(COLNMS,"^",4),?\$P(HTABS,"^",4),\$P(COLNMS,"^",5),?\$P(HTABS,"^",5),\$P(COLNMS,"^",6),?\$P(HTABS,"^",6),?\$P(COLNMS,"^",7) .F LN=1:1 S STR=\$T(@RNAME+LN) Q:STR["END OF RECORD" D ; READ \$TEXT DESCRIPTOR ..W !,\$P(STR,";",3),?\$P(FTABS,"^",1),\$P(STR,";",10),?\$P(FTABS,"^",2),\$P(STR,";",8),?\$P(FTABS,"^",3),?\$P(STR,";",7),?\$P(FTABS,"^",4),?\$P(STR,";",6),?\$P(FTABS,"^",5),?\$P(STR,";",4) U TMPIO D CLOSEFILE^CHMXWBUT(DOCFILE) ; CLOSE CURRENT FILE Q  ..... ; EMDEONHDR: Common Header for EMDEON STATUS Files ; ; A single header is generated for each output file. ; ; DESC: "FIELD NAME";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA TYPE"; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR(record # and text description) ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT, C= CENTER ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH (ANY PRINTABLE CHARACTER) ; ; NOTE: PAD CHAR="" IF NO CHARACTER IS BETWEEN THE SEMICOLONS (I.E. ;;) ; ; NO PADDING WILL OCCUR IF THIS IS SET UP THIS WAY ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; ; ; ; ; FORMATDATA TREATS THE PAD CHAR (;;) AS A NULL, SO NO PADDING OCCURS. ; ; THIS WILL ALLOW USE OF THE FORMATDATA FUNCTION WITHOUT MODIFICATION BETWEEN ; ; PADDED AND NON-PADDED FIELDS. ; ..... </pre>	

Routine Name	CHMXWB21
<pre> ; 8/4/11 DLB "6. CREATION TIME" FROM \$(DATESTAMP,9,12) TO \$(DATESTAMP,9,14) ; ; 8/15/11 DLB "2. FILE GROUP ID" INSERTED THE DATESTAMP VALUE TO ENSURE UNIQUENESS ; ; 9/7/2011 DLB 12. LOAD TYPE CHANGED TO PROVIDE "F" WHEN HISTORICAL FILE GENERATED ; ***** EMDEONHDR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";DATA LENGTH/PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"HDR";3;L;;3AN;0;R; ;;2.FILE GROUP ID;\$(GROUPID="'":GROUPID,1:DATESTAMP);20;L;;20AN;4;R; ;;3.FILE GROUP SEQUENCE NUMBER;FILECNT;2;R;0;3N;24;R; ;;4.FILE GROUP COUNT;GRPCNT;2;R;0;3N;26;R; ;;5.CREATION DATE;\$(DATESTAMP,1,8);8;L;;8AN;28;R; ;;6.CREATION TIME;\$(DATESTAMP,9,14);6;L;;6N;36;R; ;;7.TRADING PARTNER ID;"VAFNH";10;L;;10AN;42;R; ;;8.SUBMITTER NAME;"VA HEALTH ADMIN CENTER";30;L;;30AN;53;R; ;;9.PAYER CONTACT NAME;PAYNAME;60;L;;60AN;83;O; ;;10.PAYER SUPPORT TELEPHONE NUMBER;PAYPHONE;10;L;;10N;143;O; ;;11.PAYER SUPPORT EMAIL ADDRESS;"";80;L;;80AN;153;O; ;;12.LOAD TYPE;LOADTYPE;1;L;;1AN;233;R; ;;13.PAYER UNIQUE FILE IDENTIFIER;DATESTAMP;20;L;;20AN;234;R; ;;14.FILE TYPE;"CStat";5;L;;5AN;254;R; ;;15.VERSION CODE;"03";2;L;;2AN;258;R; ;;16.RELEASE CODE;"00";2;L;;2AN;260;R; ;;18.END OF RECORD;  ***** ; EMDEON CLAIM IEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for E01 File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; This table is used as a block of data(a record) for each claim to be processed. ; ; Therefore, separating this from the header and trailer tables allows multiple ; ; calls to this table. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ***** ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP1) ; ; 8/5/11 DLB FIELDS 12,14,15: TYPO FIXED FOR PROVIDER INFO (ADDED ^ TO CHMXCLB) ; ; 8/5/11 DLB FIELD 41: CHANGED INDEX FROM ^CHMXCLE(I,100 TO ^CHMXCLE(I,0 ; ***** EMDEONCLM ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA LENGTH/PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"CLM";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; </pre>	

Routine Name	CHMXWB21
<b>;;4.MAINTENANCE TYPE</b> <b>CODE;\$\$(\$G(RUNTYPE)="F": "021",\$G(RUNTYPE)="A": "021",\$G(RUNTYPE)="P": "021",\$G(RUNTYPE)="H": "030",1: "001");3;L;;3AN;19;R;</b> ;;5.BILLING PROVIDER FEDERAL TAX ID;BPFTI;9;L;;9N;22;C; ;;6.BILLING PROVIDER PAYER NUMBER;BPPN;50;L;;50AN;31;C; ;;7.BILLING PROVIDER NATIONAL ID;BPNID;10;L;;10N;81;C; ;;8.BILLING PROVIDER LAST NAME;BPLNAME;60;L;;60AN;91;C; ;;9.BILLING PROVIDER FIRST NAME;BPFNAME;35;L;;35AN;151;O; ;;10.BILLING PROVIDER MIDDLE NAME;BPMNAME;25;L;;25AN;186;O; ;;11.BILLING PROVIDER NAME SUFFIX;BPNAMEX;10;L;;10AN;211;O; ;;12.SERVICE PROVIDER FEDERAL TAX ID;SPFTID;9;L;;9N;221;C; ;;13.SERVICE PROVIDER PAYER NUMBER;SPPN;50;L;;50AN;230;C; ;;14.SERVICE PROVIDER NATIONAL ID;SPNID;10;L;;10N;280;C; ;;15.SERVICE PROVIDER LAST NAME;SPLNAME;60;L;;60AN;290;C; ;;16.SERVICE PROVIDER FIRST NAME;SPFNAME;35;L;;35AN;350;O; ;;17.SERVICE PROVIDER MIDDLE NAME;SPMNAME;25;L;;25AN;385;O; ;;18.SERVICE PROVIDER NAME SUFFIX;SPNAMEX;10;L;;10AN;410;O; ;;19.EMPLOYER IDENTIFICATION NUMBER;EMPIDNUM;80;L;;80AN;420;C; ;;20.EMPLOYER NAME;EMPNAME;60;L;;60AN;500;C; ;;21.SUBSCRIBER ID;SUBSCID;80;L;;80AN;580;C; ;;22.SUBSCRIBER LAST NAME;SUBLNAME;60;L;;60AN;660;C; ;;23.SUBSCRIBER FIRST NAME;SUBFNAME;35;L;;35AN;720;O; ;;24.SUBSCRIBER MIDDLE NAME;SUBMNAME;25;L;;25AN;755;O; ;;25.SUBSCRIBER NAME SUFFIX;SUBNAMEX;10;L;;10AN;780;O; ;;26.PATIENT ID;PATID;80;L;;80AN;790;O; ;;27.PATIENT LAST NAME;PATLNAME;60;L;;60AN;870;R; ;;28.PATIENT FIRST NAME;PATFNAME;35;L;;35AN;930;R; ;;29.PATIENT MIDDLE NAME;PATMNAME;25;L;;25AN;965;O; ;;30.PATIENT NAME SUFFIX;PATNAMEX;10;L;;10AN;990;O; ;;31.PATIENT DATE OF BIRTH;PATDOB;8;L;;8DT;1000;R; ;;32.PATIENT GENDER;PATGENDR;1;L;;1AN;1008;O; ;;33.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;1009;O; ;;34.CLAIM CHARGE AMOUNT;CLMCHRG;18;L;;18N;1059;R; ;;35.CLAIM PYMT AMT;CLMPMT;18;R;;18N;1077;C; ;;36.CLAIM ADJ/PAYMENT DATE;CAPD;8;L;;8DT;1095;C; ;;37.CHECK/EFT DATE;CHKEFTDATE;8;L;;8DT;1103;O; ;;38.CHECK/EFT NUMBER;CHKEFTNUM;16;L;;16AN;1111;O; ;;39.BILL TYPE;BTYPE;3;L;;3AN;1127;O; ;;40.PAYER CLAIM ID NUMBER;PCIDNUM;50;L;;50AN;1130;R; ;;41.PATIENT ACCOUNT NUMBER;PATACT;50;L;;50AN;1180;O; ;;42.PHARMACY PRESCRIPTION NUMBER;PRSCNUM;50;L;;50AN;1230;O; ;;43.VOUCHER IDENTIFIER;VOUCHID;50;L;;50AN;1280;O; ;;44.APP/LOCATION SYSTEM ID;LOCSYSID;50;L;;50AN;1330;O; ;;45.GROUP NUMBER;GRPNUM;50;L;;50AN;1380;O; ;;46.CLAIM SERVICE DATE START;CLMSTDT;8;L;;8DT;1430;R; ;;47.CLAIM SERVICE DATE END;CLMENDT;8;L;;8DT;1438;R; ;;48.END OF RECORD;  ..... ; EMDEON CLAIM STATUS RECORD DETAIL: Contains the Descriptions for all data to be ; ; gathered for Claim level and Line Level records. Used by \$TEXT to gather all ; ; data & set up format. ; ; Each record is described in detail for the following Parameters: ; ; NOTE: ; ; For each CLM record, there must be at least one STC record. ;	

Routine Name	CHMXWB21
	<pre> ; For a single CLM record, The following is true about DTL records: ; ; 1) Payer can send any number of DTL records (including NONE) ; ; 3) Each DTL record sent corresponds to Line Level info for the CLM record. ; ; 4) For each DTL record sent, there must be at least one STC record. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----; ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP1) ; ;-----;  EMDEONSTC ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"STC";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;3;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;13;R; ;;4.PAYER CLAIM ID NUMBER;PCINUM;50;L;;50AN;18;R; ;;5.LINE ITEM CONTROL NUMBER;LICNUM;50;L;;50AN;68;O; ;;6.STATUS INFORMATION EFF. DATE;DATESTAMP;14;L;;14N;118;R; ;;7.CLAIM STATUS CATEGORY CODE;RJCODE;3;L;;3AN;126;R; ;;8.CLAIM STATUS CODE;RJSTATUS;3;L;;3AN;129;R; ;;9.ENTITY CODE;ENTITY;3;L;;3AN;132;O; ;;10.DATA IN ERROR;DATERR;264;L;;264AN;135;O; ;;11.EMDEON STATUS CODE;ESCODE;5;L;;5AN;399;O; ;;13.END OF RECORD;  ;-----; ; EMDEONTRLR: Common Trailer for EMDEON Status Files ; ; NOTE: a single trailer record is generated for each output file. ; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: 277 File FIELD DESCRIPTOR ; ; LENGTH: 277 FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----;  EMDEONTRLR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"TRLR";4;L;;4AN;0;R; ;;2.RECORD COUNT;COUNT;10;R;;10N;4;R; ;;3.END OF RECORD;  ;-----; ; EMDEON LINE LEVEL STATUS TABLE: Contains the Descriptions for all data to be ; </pre>

Routine Name	CHMXWB21
	<pre> ; gathered for CRF File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; NOTE: there may be multiple line items for each claim record. This table ; ; defines the record to be generated for each line item. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----; EMDEONLI ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"DTL";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.PAYER CLAIM ID NUMBER;PCID;50;L;;50AN;19;R; ;;5.LINE ITEM CONTROL NUMBER;LICN;50;L;;50AN;69;R; ;;6.SERVICE QUALIFIER ID;SQID;2;L;;2AN;119;R; ;;7.SERVICE IDENTIFICATION CODE;SICOD;48;L;;48AN;121;R; ;;8.PROCEDURE MODIFIER 1;PRCM1;2;L;;2AN;131;O; ;;9.PROCEDURE MODIFIER 2;PRCM2;2;L;;2AN;133;O; ;;10.PROCEDURE MODIFIER 3;PRCM3;2;L;;2AN;135;O; ;;11.PROCEDURE MODIFIER 4;PRCM4;2;L;;2AN;137;O; ;;12.LI CHARGE AMOUNT;LICHARGA;18;L;;18N;139;R; ;;13.LINE ITEM PROV. PAYMENT AMOUNT;LIPPA;18;L;;18N;157;R; ;;14.REVENUE CODE;RVNUCODE;48;L;;48AN;175;C; ;;15.QUANTITY(UNITS OF SERVICE);QTYUOS;15;L;;15N;223;O; ;;16.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;238;O; ;;17.SRVC START DATE;SVCSTDATE;8;L;;8N;288;R; ;;18.SRVC END DATE;SVCENDATE;8;L;;8N;306;R; ;;20.END OF RECORD; </pre>

### 3.2.1.2 Archive All Final Claim Status Files for Primary Claims

#### Functional Requirement: 2.8.1.2 Save Final Claim Status Files – Primary Claims

The CP&E system shall provide persistent storage of the Final Claim Status files for electronic primary CHAMPVA/SB/CWVV claims after sending them to the HCCH. The system will use existing methodology to perform the Archive process. The following routine places the file(s) in the proper VMS directory for pickup by the FTP MOVEit process.

Routine Name	<b>CHMXWB21</b>		
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change		
Requirement Traceability Matrix	2.8.1.2		
Related Options			
Related Routines	Routines "Called By"	Routines "Called"	
	Taskman process	CHMXWBUT	
Data Dictionary (DD) References			
Related Protocols			
Related Integration Control Registrations (ICRs)			
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local		
Input Attribute Name and Definition	Name: Definition:		
Output Attribute Name and Definition	Name: Definition:		
Current Logic			
Same as 3.2.1.1			
Modified Logic (Changes are in bold and highlighted)			
<pre> GETFILE()   N DIRPATH   S EICNT=0,FILECNT=FILECNT+1   <b>S DIRPATH="CHAMPVA EDI:[KERMIT.WEBMD]"</b>   HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP   D NOW^%DTC   S FMDATE=%   S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")   S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")   S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND   <b>I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT"</b>   <b>E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT"</b>   ; CHMEDCOB SET IN CHMXWT1   E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME   I RUNTYPE="H" S DIRFILE=DIRFILE_"_H"   I RUNTYPE="O" S DIRFILE=DIRFILE_"_O"   S FLAG=\$\$OFFILE^CHMXWBUT(DIRFILE,"NWS")   I 'FLAG D VMSERR Q ""   S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q </pre>			

### 3.2.1.3 Transmit Daily Final Claim Status Files for Primary Claims

#### Functional Requirement: 2.8.1.3 Transmit Daily Final Claim Status File – Primary Claims

The CP&E system shall transmit final claim statuses for electronic primary CHAMPVA/SB/CWVV claims to the HCCH daily via FTP upon project implementation. The FTP process will use existing methodology. The following routine places the file(s) in the proper VMS directory for pickup by the FTP MOVEit process.

Routine Name	<b>CHMXWB21</b>		
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change		
Requirement Traceability Matrix	2.8.1.3		
Related Options			
Related Routines	Routines "Called By"	Routines "Called"	
	Taskman process	CHMXWBUT	
Data Dictionary (DD) References			
Related Protocols			
Related Integration Control Registrations (ICRs)			
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local		
Input Attribute Name and Definition	Name: Definition:		
Output Attribute Name and Definition	Name: Definition:		
Current Logic			
Same as 3.2.1.1			
Modified Logic (Changes are in bold and highlighted)	<pre> GETFILE()   N DIRPATH   S EICNT=0,FILECNT=FILECNT+1   <b>S DIRPATH="CHAMPVA EDI:[KERMIT.WEBMD]"</b>   HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP   D NOW^%DTC   S FMDATE=%   S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")   S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")   S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND <b>I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT"</b> </pre>		

Routine Name	CHMXWB21
<b>E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT"</b> ; CHMEDCOB SET IN CHMXWT1 E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME I RUNTYPE="H" S DIRFILE=DIRFILE_"_H" I RUNTYPE="O" S DIRFILE=DIRFILE_"_O" S FLAG=\$\$OFIL^CHMXWBUT(DIRFILE,"NWS") I 'FLAG D VMSERR Q "" S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q	

### 3.2.1.4 Create Final Claim Status File Data for Primary Claims

#### Functional Requirement: 2.8.1.4 Final Claim Status File Data – Primary Claims

The CP&E system shall create the Final Claim Status File(s) based on the Emdeon / Hosted Real-Time Specification document and adhere to all file specifications and limitations. The specified file size limit of 500 mb per file will be modified to improve submission performance. The new file size limit will be 150 mb. The CP&E system shall transmit the following data to the HCCH in the Final Claim Status file for electronic primary CHAMPVA/SB/CWVV claims when available:

- Header: Record ID = HDR - Required
- Header: File Group ID - Required
- Header: File Group Sequence Number - Required
- Header: File Group Count - Required
- Header: Creation Date - Required
- Header Creation Time - Required
- Header: Trading Partner ID = VAFNH - Required
- Header: Submitter Name = VA Health Administration Center - Required
- Header: Payer Contact Name = CHAMPVA Customer Service - Required
- Header: Payer Support Telephone Number = 1-800-733-8387 (CHAMPVA), 1-888-820-1756 (SB/CWVV) - Required
- Header: Payer Support Email Address - Optional
- Header: Load Type = I - Required
- Header: Payer Unique File Identifier - Required
- Header: File Type = CStat - Required
- Header: Version Code = 03 - Required
- Header: Release Code = 00 - Required
- Header: Record Terminator = CRLF - Required
- Claim Level Record Details: Record ID = CLM - Required
- Claim Level Record Details: Record Number = 2 and increments by 1 - Required
- Claim Level Record Details: Payer ID = VAHAC - Required
- Claim Level Record Details: Maintenance Type Code = 021 - Required



- Claim Level Record Details: Billing Provider Federal Tax ID - Conditional
  - Required when the Billing Provider Payer Assigned Number and the Billing Provider National Provider ID are not present
- Claim Level Record Details: Billing Provider Payer Assigned Number - Conditional
  - Required when the Billing Provider Federal Tax ID and the Billing Provider National Provider ID are not present
- Claim Level Record Details: Billing Provider National Provider ID - Conditional
  - Required when the Billing Provider Federal Tax ID and the Billing Provider Payer Assigned Number are not present
- Claim Level Record Details: Billing Provider Last Name/Organization Name - Optional
- Claim Level Record Details: Billing Provider First Name - Optional
- Claim Level Record Details: Billing Provider Middle Name - Optional
- Claim Level Record Details: Billing Provider Name Suffix - Optional
- Claim Level Record Details: Service Provider Federal Tax ID - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, must be either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID, else not used
- Claim Level Record Details: Service Provider Payer Assigned Number - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, must be either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID, else not used
- Claim Level Record Details: Service Provider National Provider ID - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, must be either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID, else not used
- Claim Level Record Details: Service Provider Last Name/Organization Name - Conditional
  - Only used if either Service Provider Federal Tax ID Number, Service Provider Payer Assigned Number, or the Service Provider National Provider ID is present
- Claim Level Record Details: Service Provider First Name - Optional
- Claim Level Record Details: Service Provider Middle Name - Optional
- Claim Level Record Details: Service Provider Name Suffix - Optional
- Claim Level Record Details: Employer Identification Number - Conditional
  - Required if Employer Name is present
- Claim Level Record Details: Employer Name - Conditional
  - Used in cases of Employer-Subscriber
- Claim Level Record Details: Subscriber ID - Conditional
  - Required when Subscriber Last Name is present
- Claim Level Record Details: Subscriber Last Name - Conditional
  - Used when the Subscriber is an insured individual

- Claim Level Record Details: Subscriber First Name - Optional
- Claim Level Record Details: Subscriber Middle Name - Optional
- Claim Level Record Details: Subscriber Name Suffix - Optional
- Claim Level Record Details: Patient ID - Optional
- Claim Level Record Details: Patient Last Name - Required
- Claim Level Record Details: Patient First Name - Required
- Claim Level Record Details: Patient Middle Name - Optional
- Claim Level Record Details: Patient Name Suffix - Optional
- Claim Level Record Details: Patient Date of Birth - Required
- Claim Level Record Details: Patient Gender - Optional
- Claim Level Record Details: HCCH Claim Number - Required
- Claim Level Record Details: Claim Charge Amount - Required
- Claim Level Record Details: Claim Payment Amount - Conditional
  - Required when Claim Status Category Code 1-3 = F1
- Claim Level Record Details: Claim Adjudication/Payment Date - Conditional
  - Required when first position of Claim Status Category Code = F
- Claim Level Record Details: Check/EFT Date - Optional
- Claim Level Record Details: Check/EFT Number - Optional
- Claim Level Record Details: Bill Type - Optional
- Claim Level Record Details: Payer Claim Identification Number - Required
- Claim Level Record Details: Patient Account Number - Required when provider claim number received
- Claim Level Record Details: Pharmacy Prescription Number - Optional
- Claim Level Record Details: Voucher Identifier - Optional
- Claim Level Record Details: Application or Location System Identifier - Optional
- Claim Level Record Details: Group Number - Optional
- Claim Level Record Details: Claim Service Date Start - Required
- Claim Level Record Details: Claim Service Date End - Required
- Claim Level Record Details: Record Terminator = CRLF - Required
- Claim Line Level Record Details: Record ID = DTL - Required
- Claim Line Level Record Details: Record Number - Required
- Claim Line Level Record Details: Payer ID - Required
- Claim Line Level Record Details: Payer Claim Identification Number - Required
- Claim Line Level Record Details: Line Item Control Number - Required
- Claim Line Level Record Details: Service Qualifier ID - Required
- Claim Line Level Record Details: Service Identification Code - Required
- Claim Line Level Record Details: Procedure Modifier 1 - Optional
- Claim Line Level Record Details: Procedure Modifier 2 - Optional
- Claim Line Level Record Details: Procedure Modifier 3 - Optional
- Claim Line Level Record Details: Procedure Modifier 4 - Optional

- Claim Line Level Record Details: Line Item Charge Amount - Required
- Claim Line Level Record Details: Line Item Provider Payment Amount - Required
- Claim Line Level Record Details: Revenue Code - Optional
- Claim Line Level Record Details: Quantity (Units of Service) - Optional
- Claim Line Level Record Details: HCCH Claim Number - Required
- Claim Line Level Record Details: Date of Service Start - Required
- Claim Line Level Record Details: Date of Service End - Required
- Claim Line Level Record Details: Record Terminator = CRLF - Required
- Claim Status Record Detail: Record ID = STC - Required
- Claim Status Record Detail: Record Number - Required
- Claim Status Record Detail: Payer ID - Required
- Claim Status Record Detail: Payer Claim Identification Number - Required
- Claim Status Record Detail: Line Item Control Number - Required
- Claim Status Record Detail: Status Information Effective Date - Required
- Claim Status Record Detail: Claim Status Category Code - Required
- Claim Status Record Detail: Claim Status Code - Required
- Claim Status Record Detail: Entity Code - Optional
- Claim Status Record Detail: Data in Error - Optional
- Claim Status Record Detail: HCCH Status Code - Optional
- Claim Status Record Detail: Record Terminator = CRLF - Required
- Trailer Record Details: Record ID = TRLR - Required
- Trailer Record Details: Record Count - Required
- Trailer Record Details: Record Terminator = CRLF - Required

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.1.4	
Related Options		
<b>Related Routines</b>	<b>Routines “Called By”</b>	<b>Routines “Called”</b>
	<b>Taskman process</b>	<b>CHMXWBUT</b>
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	

Routine Name	<b>CHMXWB21</b>
Input Attribute Name and Definition	Name: Definition:
Output Attribute Name and Definition	Name: Definition:
Current Logic	
Same as 3.2.1.1	
Modified Logic (Changes are in bold and highlighted)	
<pre> CHMXWB21 ;HAC/DLB;WEB 5010 277 UNSOLICITED STATUS/DRIVER PART 1;10/25/10 ;;1.0;CHAMPVA SYSTEM;;JULY 11,2011;Build 1 ;;V1.0; ;;CALLED BY ^CHMXG001 TO PERFORM THE "ACK" STATUS ;;CALLED BY TASKMAN TO PERFORM THE "PENDING" AND "FINAL" STATUS ;;CALLS ENTRY POINTS TO CHMXWB22,CHMXWBUT ;;HR-COB-Medicare-A/B-Begin-9372 (27-May-2010)  ** ** ** 10/18/2011 ADDED TSTBLDCLM() ENTRY POINT TO VALIDATE THE REJECT VALUES BEING ** REPORTED IN THE CSTAT FILE ** 10/20/2011 MODIFIED THE LINE() FUNCTION TO FIX INCORRECT REJECT STATUS ** REPORTING IN THE CSTAT FILE ** 10/21/2011 ADDED THE SIX DATE GATHERING FUNCTIONS TO SUPPORT THE RULE ** PROVIDED BY BUSINESS GROUP FOR I/P/D CLAIMS ** 10/21/2011 ADDED THE GETCLMTYP() FUNCTION TO DETERMINE THE I/P/D CLAIM TYPE ** FOR BOTH 4010 AND 5010 INCOMING CLAIMS ** 10/24/2011 CHANGED THE DATERANGE FUNCTION TO A EXISTING DATE FUNCTION ** IN THE UTILITY ROUTINE: CHMXWBUT.INT ** 10/25/2011 CHANGED THE RENDERING PHYSICIAN NPI RETRIEVAL TO GET THE DATA ** FROM THE CHMXCLE(I,64) TIN DATA FIELD. ** 10/25/2011 ADDED A TEST FOR THE "U" VALUE RECEIVED FROM EMDEON TO RETURN "" ** IF NOT A "M" OR "F" (ONLY ACCEPTED VALUES) ** 10/26/2011 REMOVED ERROR MESSAGE OUTPUT FOR ERROR TRAP HANDLER ** ** ** 10/26/2011 CHANGED THE INSTITUTIONAL AND DENTAL START/END DATES FOR CLAIM ** LEVEL REPORTING TO GET START FROM ^CHMXCLE(I,1),FIELD 1 AND END ** TO ^CHMXCLE(I,1) FIELD 2 ** 11/4/2011 CHANGED THE DATE CHECK ROUTINE GETCLMI() TO RETURN THE CORRECT ** DATE FOR THE DESIRED FLAG ** ** ;1.0 IS THE INITIAL VERSION (HIPAA Ready LLC) ;2.0 UPDATES TO THE STATUS REPORT PER HIPAA II SPECIFICATIONS. ; *** Filenames should not be hard coded in files. See ^CHMX277 TRACKING ; GLOBAL FILE for the destination directory/filenames for the 5010 status ; files. ; Q  ***** ; PRODUCTION ENTRY POINTS TO PROCESS THE CURRENT STATUS. ; *****  EPACK(CHMXI) ; SET UP THE FILE/BATCH INDEX (PROVIDED BY CHMXG001) ; CHMXI I INDEX INTO ^CHMXCL() N COUNT,DIRFILE,GRPCNT,FILECNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA </pre>	

Routine Name	CHMXWB21
<p>TE,CHTPI,CHTPARR,CNT  N MAXCNT,EICNT,CHA,CHMEDCOB,TPINT  S FILECNT=0,RUNTYPE="A",GRPCNT=1,EICNT=0  S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER  D INIT  S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA=""  S CHTPI=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:CHTPI="" ; GET CLAIM T/P FOR FILE  BEING CHECKED  S CHTPI=\$O(^CHMXTP("C",CHTPI,"")) Q:CHTPI="" D ; GET "LOCAL" TP ID  S IDX=\$O(^CHMX277("B",CHTPI,0))  S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB"  EMDEON PARTNER  I CHMEDCOB=0 S TPINT=\$P(^CHMX277(IDX,0),"^")  I TPINT="" S CHMEDCOB=+\$P(\$G(^CHMXTP(TPINT,0)),"^",11)  D GETCHCLEI(CHMXI,CHTPI) ; RETRIEVE ^CHMXCLE INDEX  Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE  D CREATEFILE  D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION  Q</p> <p>EPPENDTST  N TEST S TEST=1  D EPPEND  Q</p> <p>EPPEND  N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA  TE,CHTPI,CHTPARR  N CHNODEDEF,BREF,CHIIKVAL,DRDATA,TRKDATA,MAXCNT,CHMEDCOB,IDX  ; FILEMAN UPDATE VARIABLES  N CHTO,CHFROM,TRKI,EICNT  S RUNTYPE="P",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE  D INIT  S CHTPI="A"  F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM  "B" XREF  .S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER  .S IDX=\$O(^CHMX277("B",CHTPI,0))  .S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB"  EMDEON PARTNER  .K CHTPARR  .S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY  .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX  .Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0)  .S CHTPARR(CHTPI)="" ; SET CHTPARR UP FOR  GETPENDEI  .D GETTOFROM(CHTPI) Q:(CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL  .D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES  .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE  .D CREATEFILE ; CREATE THE  PENDING STATUS FILE  .D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION  Q</p> <p>EPFINAL ; entry point to start Final Status Creation file</p>	

Routine Name	CHMXWB21
	<pre> N RUNTYPE,GRPCNT,EICNT,GROUPID,CHTPI,CHTPARR,CHFROM,CHTO,FILECNT N TRKI N COUNT,DIRFILE,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,FMDATE N CHNODEDEF,BREF,CHIIKVAL,DRDATA,TRKDATA,MAXCNT          ; FILEMAN UPDATE VARIABLES S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D          ;TRADING PARTNERS FROM "B" XREF .K CHTPARR S (CHFROM,CHTO,FILECNT,CHMEDCOB)=0 ; KILL TRADING PARTNER ARRAY; ASSUME NOT A "COB" TRADING PARTNER .S TRKI=\$O(^CHMX277("B",CHTPI,0)) ; GET TRACKING GLOBAL INDEX .I \$P(\$G(^CHMX277(TRKI,0)),"^",3)=0 Q .I \$P(^CHMX277(TRKI,0),"^",4)["COB" S CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .S CHTPARR(CHTPI)=" " .D GETTOFROM(CHTPI) Q:(CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETFINALEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE          ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPREPTST N TEST S TEST=1 D EPREPLACEMENT Q  EPREPLACEMENT N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT,CHMEDCOB,IDX,CHTPX S RUNTYPE="H",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="REPLACEMENT" S TP=\$\$LISTTP^CHMXWB22 ; SELECT TRADING PARTNER S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER S IDX=\$O(^CHMX277("B",TP,0)) S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_.999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHXI="" .F S CHXI=\$O(^CHMXCL("B",IDATE,CHXI)) Q:CHXI="" D ..S CHMXI=CHXI ..I \$D(TEST) W !,"CHECKING:",\$\$DTCVRT(\$E(IDATE,1,7)) ; SHOW FILE DATES BEING PROCESSED </pre>

Routine Name	CHMXWB21
<pre> ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S CHTPX=\$P(\$G(^CHMXCLA(CHA,1)),",",1) Q:CHTPX="" ; GET CLAIM T/P FOR FILE BEING CHECKED ..S CHTPX=\$O(^CHMXTP("C",CHTPX,"")) ; CHECK SELECTED VS CLAIM TP ID ..I TP=CHTPX D ; FILE MATCHES THE SELECTED TP ..S CHTPI=TP ; CHTPI IS USED DOWNSTREAM ...I \$D(TEST) W " PROCESSING:",\$\$DTCVRT(\$E(IDATE,1,7)) ...I \$D(TEST) W !," EPREPLACEMENT: CHMXI= ",CHMXI," CHA= ",CHA," CHTPI= ",CHTPI," TP= ",TP ..S CHTPARR(TP)=" D GETCHCLEI(CHMXI,CHTPI) ; IF TP VALID, PROCESS FILE I \$D(TEST) W !,"REPLACE: \$J= ",\$J," REC COUNT= ",\$G(^ZSC(\$J,0))," CHTPI= ",CHTPI," TP= ",TP Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  DTCVRT(DATE) ; CONVERT DATE TO EXTERNAL FORMAT N EXTDATE S EXTDATE=\$E(EXTDATE,4,5)_"-"_ \$E(EXTDATE,6,7)_"-"_(\$E(EXTDATE,1,3)+1700) Q EXTDATE  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPOVERTST N TEST S TEST=1 D EPOVERLAY Q  EPOVERLAY N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT S RUNTYPE="O",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="OVERLAY" S TP=\$\$LISTTP^CHMXWB22 D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHMXI="" .F S CHMXI=\$O(^CHMXCL("B",IDATE,CHMXI)) Q:CHMXI="" D ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S TPTIN=\$P(\$G(^CHMXCLA(CHA,1)),",",1) Q:TPTIN="" ..I TP=\$O(^CHMXTP("C",TPTIN,"")) S CHTPARR(TP)=" D GETCHCLEI(CHMXI,TP) Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  ..... ; INIT PERFORMS LOCAL VARIABLE INITIALIZATION REQUIREMENTS FOR THE STATUS ; </pre>	

Routine Name	CHMXWB21
<pre> ; PROCESSES. ; ..... INIT ; K ^ZSC(\$J) S ^ZSC(\$J,0)=0 K TMP S GROUPID="",CHTPI=0 I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S ATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND Q  ..... ; EPSTAT IS THE ENTRY POINT FOR THE ON-DEMAND REPORT FOR STATISTICS ; .....  EPSTAT D ONDEMAND^CHMXWB22 Q  EPREGEN D REGEN^CHMXWB22 Q  ..... ; FILE CREATION FUNCTIONS ; .....  CREATEFILE ; S MAXCNT=500000 ; HAPE POR 11/21/13 force max file size under 150 meg I RUNTYPE="F" S MAXCNT=175000 D FILELOOP Q  FILELOOP ; BUILD CLAIMS FROM ^ZSC(\$J) ARRAY (^CHMXCLE(I) VALUES) N CHCLEI,PDI D GETGRPCNT ; GET ESTIMATED GROUP FILE COUNT Q:GRPCNT=0 D GETFILE U DIRFILE ; CREATE ORIGINAL FILE D BLDHDR ; BLDHDR SETS COUNT=2 (SEE EMDEON SPEC) S CHCLEI=0 F S CHCLEI=\$O(^ZSC(\$J,CHTPI,CHCLEI)) Q:CHCLEI="" D .S EICNT=EICNT+1 .S STVAL=\$G(^ZSC(\$J,CHTPI,CHCLEI)) .D BLDCLM(CHCLEI,STVAL) ; GENERATE CLAIM/STC,DTL/STC RECORDS .D CHKSEQ ; CHECK FOR 500MB LIMIT, CREATE NEW FILE D BLDTRL ; BUILD THE TRAILER RECORD D CLOSEFILE ; CLOSE CURRENT FILE </pre>	



Routine Name	CHMXWB21
<p><b>Q</b></p> <p>GETGRPCNT ; ESTIMATE THE TOTAL NUMBER OF FILES REQUIRED  N RECCNT,X  S RECCNT=0  S X=""  F S X=\$O(^ZSC(\$J,CHTPI,X)) Q:X="" S RECCNT=RECCNT+1  S GRPCNT=RECCNT\MAXCNT ; INTEGER DIVISION OF RECORD COUNT  S:RECCNT#MAXCNT&gt;0 GRPCNT=GRPCNT+1  ;W !,"GRPCNT= ",GRPCNT  Q</p> <p>CHKSEQ ; HANDLE THE FILE SEQUENCE NUMBERING REQUIREMENTS  I EICNT&gt;MAXCNT D  .D BLDTRL ; OUTPUT TRAILER RECORD  .D CLOSEFILE  .D GETFILE U DIRFILE  .D BLDHDR ; BLDHDR RESETS "COUNT" VARIABLE  Q</p> <p>FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN  Q:"APF"[RUNTYPE ; A=ACK, P=PENDING, F=FINAL  N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA  S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_COUNT_"^.05="_CHMXI  T_"^.05="_CHMXI  S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_CHTO_"^.05="_CHFROM_"^.06="_COUNT  _"^.05="_CHFROM_"^.06="_COUNT  S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_CHTO_"^.05="_CHFROM_"^.06="_COUNT  _"^.05="_CHFROM_"^.06="_COUNT  S BREF=FMDATE  S CHIKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE  S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT  D ADD^CHHRLIBFM(CHNODEDEF,CHIKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL  I RUNTYPE="F" N X,X1 F S X=\$O(^ZSC1(\$J,X)) Q:X="" F S X1=\$O(^ZSC1(\$J,X,X1)) Q:X1="" I \$D(^CHMPAY("E",X,X1)) S ^CHMPAY("E",X,X1)=1  Q</p> <p>.....  ;TEST ENTRY POINT TO TEST THE BLDCLM FUNCTION FROM A PDI VALUE  ; THERE ARE MULTIPLE CROSS REFERENCES (SOME DOCUMENTED, SOME UNDOCUMENTED  ; THAT ARE UTILIZED IN THE CLAIM PROCESSING SYSTEM. THE CVTPDI AND TSTBLDCLM  ; FUNCTIONS UTILIZE THE MAJORITY OF THESE CROSS REFERENCES.  .....</p> <p>CVTPDI(CHREFN)  N CHPCN,CHMXCLI,CHMXID,CHCLFI,SDATE,EDATE,CHCLFI,CHCLEI,IDX  I '\$D(^CHMXCLE("PDI",CHREFN)) W !,"NO ^CHMXCLE("PDI",",CHREFN,") XREF" Q  S CHPCN=""  S1 S CHPCN=\$O(^CHMXCLE("PDI",CHREFN,CHPCN)) I CHPCN="" Q ;VALID PDI,NO DATA  S CHMXCLI=0  S2 S CHMXCLI=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI)) I 'CHMXCLI Q ;VALID PDI,NO DATA</p>	

Routine Name	CHMXWB21
S3	<pre> S CHMXID="" S CHMXID=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI,CHMXID)) I CHMXID="" Q S CHCLEI=\$P(CHMXID,"*",4) W !,"CLM SDATE [^CHMXCLE(I,1),1]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) W !,"CLM EDATE [^CHMXCLE(I,1),2]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) S CHCLFI=0 W !,"LINE ITEM DATE VALUES" F IDX=1:1 S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI D ; CHECK ALL LINE ITEM DATES .W !,IDX," ", " L/I SDATE [^CHMXCLF(I,1),11]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) .W !," L/I END [^CHMXCLF(I,1),12]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) .W !," THERAPY SDATE [^CHMXCLF(I,2.5),1]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",1) .W !," LAST CERT DATE [^CHMXCLF(I,2.5),2]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",2) Q  TSTBLDCLM(CHPDI) N CHPCN,CHCLI,CHCLAI,CHCLEI,STVAL,COUNT,PDI,BUFI,RUNTYPE,CLMTYPE,TEST S (STVAL,CHPCN)="", (COUNT,CHCLEI,CHCLAI,PDI)=0,TEST=1 S CHPCN=0,CHPCN=\$O(^CHMXCLE("PDI",CHPDI,CHPCN)) I CHPCN="" W !,"NO CHPCN VALUE" Q E W !,"PATIENT CONTROL # [^CHMXCLE("PDI",CHPDI,PCN)]= ",CHPCN ; PATIENT CONTROL NUMBER S CHCLI=0,CHCLI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI)) I CHCLI="" W !,"NO CHCLI VALUE" Q S STVAL=0,STVAL=\$O(^CHMXCLE("A",CHCLI,STVAL)) W !,"XREF CLAIM STATUS= ",STVAL S BUFI="",BUFI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI)) I BUFI="" W !,"NO BUFFER INDEX VALUES" Q W !,"XREF [\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))] BUFFER INDICES= ",BUFI S CHCLEI=\$P(BUFI,"*",4) I CHCLEI="" W !,"NO CHCLEI VALUE RECOVERED" S RUNTYPE="H" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; RETRIEVE PDI S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) W !,"PDI VALUE FROM [^CHMXCLE(I,100),2 OR 4]= ",PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),^",1) W !,"X12 837 BUFFER [CHMXCL(I)= ",?30,CHCLI W !,"TRANSACTION [CHMXCLA(I)]= ",?30,CHCLAI W !,"PROVIDER BUFFER [CHMXCLB(I)]= ",?30,CHCLBI W !,"PATIENT BUFFER [CHMXCLC(I)= ",?30,CHCLCI W !,"CLM BUFFER [CHMXCLE(I)= ",?30,CHCLEI W !,"L/I BUFFER [CHMXCLF(I)]= ",?30,\$O(^CHMXCLF("B",CHCLEI,0)) S CLMTYPE=\$GETCLMTYP W !,"VERSION-FUNCTIONAL TYPE [^CHMXCLA(I,0),13]= ", \$P(\$G(^CHMXCLA(CHCLAI,0)),^",13) W !,"CLAIM TYPE= ", \$S(CLMTYPE="I": "INSTITUTIONAL",CLMTYPE="P": "PROFESSIONAL",CLMTYPE="D": "DENTAL") ; I/P/D CLAIM W !,"STATUS (ACK/PENDING)=", \$S(\$D(^CHMPAY("C",PDI))=0: "ACK",1: "PENDING") D CVTPDI(CHPDI) W ! D BLDCLM(CHCLEI,"") Q </pre>

```

.....
; BLDCLM: ENTRY POINT TO BUILD THE CSTAT RECORDS
; THIS FUNCTION PROVIDES AN EXTERNAL ENTRY POINT TO GENERATE THE
; EMDEON CLAIM, CLAIM STATUS, DETAIL, AND DETAIL STATUS RECORDS
.....

```

**BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS**

**; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE**

**; STVAL STATUS VALUE AFTER FROMNT END EDITS**

**N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE**

**I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE**

**THIS**

**.D NOW^%DTC**

**.S FMDATE=%**

**.S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")**

**.S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")**

**.S DATESTAMP=(\$E(FMDATE,1,7)+17000000)\_TIME ; DATE/TIME FILE CREATED DOWN**

**TO SECOND**

**S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1);TRAVERSE BACK THROUGH BUFFER FILES**

**S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1);GET "I" VALUES FOR ALL BUFFER FILES**

**S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1)**

**S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1)**

**S:'\$D(COUNT) COUNT=0**

**K CHRJARR**

**S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL**

**I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1\*45\*" D ; ALL "PENDING" = P1/45**

**.D CLM**

**.D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS**

**I STYPE="A" D**

**.D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR)**

**.I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT**

**END EDIT FAIL)**

**.I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2\*20\*" ; NO FRONT END EDITS GET**

**A2/20**

**.D CLM**

**.D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS**

**.K CHRJARR S CHRJARR(0)=0**

**.S CHCLFI=0**

**.F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D**

**..D GLINRJRSN^CHMXWBUT(CHCLFI)**

**.D:CHRJARR(0)>0 LINE ; NO LINE**

**REJECTS RECORDED = NO LINE REPORTS**

**I STYPE="F" D**

**. S (CHKDATE,CHKEFT,PAYDATE)=""**

**. S CHRJARR(0)=1**

**. F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D**

**.. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1\*65"**

**.. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,'\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2\*1"**

**..**

**I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)>CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^",16)**

**.. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)>PAYDATE S PAYDATE=\$P(^0,"^",10)**

**. I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4)**

**. I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4)**

Routine Name	CHMXWB21
	<pre> . D CLM . D BLDSTC("CLM") . D LINE Q  GETSTYPER(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D . S CHMPAY=0,OK=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)),^",2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)),^",2)=0 Q .. S OK=0 .. Q Q OK  ..... ; FUNCTIONS TO GATHER INFORMATION REGARDING REJECT REASONS ; .....  BLDACK ; POPULATE CHRJARR REJECT REASON/REJECT CODE VALUES K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY S REJCODES="" D GTRXRJRSN^CHMXWBUT(CHCLAI) ; TRANSACTION "101" NODE REJECT REASONS D GPRORJRSN^CHMXWBUT(CHCLBI) ; PROVIDER "101" NODE REJECT REASONS D GPATRJRSN^CHMXWBUT(CHCLCI) ; PATIENT "101" NODE REJECT REASONS D GCLMRJRSN^CHMXWBUT(CHCLEI) ; CLAIM "101" NODE REJECT REASONS I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" ;I \$D(TEST) W !,"BLDACK: RJCNT=",CHRJARR(0)," RJARR(1,1)=",CHRJARR(1,1) Q  ..... ; RECORD BUILDING FUNCTIONS: \$TEXT FILES PROVIDE FIELD DEFINITIONS ; .....  BLDHDR ;BUILD HEADER RECORD N LN,REC,STR,LOADTYPE S (STR,LN,REC)="",COUNT=2 ; EMDEON SPEC: REC. COUNT STARTS @ 2 D GETHDRVAL ; SET UP HEADER VARIABLES F LN=1:1 S STR=\$T(EMDEONHDR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="" Q  CLM ; </pre>

Routine Name	CHMXWB21
<pre> N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDY,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q  LINE ; BUILD DTL RECORDS FOR LINE ITEMS N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDY,SVCENDY N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D .D GDTLVARS .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),^",) .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1" Q ... S CHRJARR(1,1)="F0*1" ... Q .. Q .D BLDSTC("DTL") Q  BLDSTC(TYPE) ; WRITE OUT THE STC RECORDS FROM CHRJARR REJECT ARRAY ; TYPE CLAIM OR DETAIL LEVEL EMDEON STATUS VALUES &amp; LINE ITEM CONTROL NUMBER N PCINUM,LICNUM,DATERR,ESCODE N IDX,JDX,REJCODES,ECNT N LN,STR,REC S REJCODES=0 </pre>	

Routine Name	CHMXWB21
<pre> S ECNT=CHRJARR(0) ; FRONT END EDIT REJECT COUNT F IDX=1:1 Q:(IDX&gt;ECNT) D          ; PROVIDE STC RECORD FOR EACH ERROR ..F JDX=1:1 Q:\$G(CHRJARR(IDX,JDX))="" D ; GET EACH REJECT VALUE STORED ..S (RJCODE,RJSTATUS,ENTITY)="" ..D GETSTCVAL(TYPE) ; EMDEON STATUS RECORD VALUES FOR CLM OR DTL ..S REJCODES=\$G(CHRJARR(IDX,JDX)) ..;W:\$D(TEST) !,"BLDSTC: REJCODE STR=",REJCODES,! ; DEBUG: ONLY OUTPUTS IF "TEST" IS DEFINED ..S RJCODE=\$P(REJCODES,"*",1),RJSTATUS=\$P(REJCODES,"*",2),ENTITY=\$P(REJCODE S,"*",3) ..S REC="" ..F LN=1:1 S STR=\$T(EMDEONSTC+LN) Q:STR["END OF RECORD" D          ; GENERATE THE STATUS (STC) RECORD ...I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ...E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 Q  BLDTRL                                     ; TRAILER RECORD N LN,REC,STR S COUNT=COUNT-2 ; ADJUST COUNT FOR START OF 2 (SEE EMDEON SPEC) S (STR,LN,REC)="" F LN=1:1 S STR=\$T(EMDEONTRLR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ; 10/20/2010 ADDED "ELSE" W REC,! S REC="" Q  ..... ; GENERIC FUNCTIONS SUPPORTING THE RECORD BUILDING ; .....  GETFILE() N DIRPATH S EICNT=0,FILECNT=FILECNT+1 S DIRPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP D NOW^%DTC S FMDATE=% S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT" E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT" ; CHMEDCOB SET IN CHMXWT1 E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME I RUNTYPE="H" S DIRFILE=DIRFILE_"_H" I RUNTYPE="O" S DIRFILE=DIRFILE_"_O" S FLAG=\$\$OFILE^CHMXWBUT(DIRFILE,"NWS") I 'FLAG D VMSERR Q "" S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q </pre>	

Routine Name	CHMXWB21
<pre> TST     N FMDATE,TIME     S FMDATE=3110901.1234     W !,FMDATE     S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")     W !,FMDATE     S TIME=\$E(FMDATE,9,14)     W !,TIME     S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")     W !,TIME     Q  CLOSEFILE     C DIRFILE     Q  GETCHCLEI(CHCLI,CHTPI) ; BUILD ARRAY OF CLAIM INDICES (CHMXCLE(I) VALUES)     ; CHCLI I INDEX FOR ^CHMXCL() FILE (CLAIM FILE RECEIVED)     N STAT,CHCLAI,PDI,CHIDXS,CHCLEI,RECCNT,CTPI S CTPI=0     S STAT="",STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) I \$D(TEST) W:STAT="" !,"NULL STAT" Q:STAT="" ; RETRIEVE STATUS VALUE     S CHCLAI="",CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) I \$D(TEST) W:CHCLAI="" !,"NULL CHCLAI" Q:CHCLAI="" ; RETRIEVE THE ^CHMXCLA(I) VALUE     I \$D(TEST) W !," GETCHCLEI: READING:",CHCLI," STATUS= ",STAT," ^CHMXCLA(I)= ",CHCLAI," TP= ",CTPI     S STAT=""     F S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) Q:STAT="" D     .S CHCLAI=""     .F S CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) Q:CHCLAI="" D     ..S PDI=0     ..F S PDI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI)) Q:'PDI D     ...S CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,0))     ...Q:CHIDXS="" ; PDI RETRIEVES BUFFER IDXs     ...S CHCLEI=\$P(CHIDXS,"*",3)     ...S ^ZSC(\$J,CHTPI,CHCLEI)=STAT,RECCNT=\$I(^ZSC(\$J,0))     Q  GETTP(CHA)     N TPID,ID     S ID=""     S TPID=\$P(\$G(^CHMXCLA(CHA,1)), "^",1)     S:TPID]"" ID=\$O(^CHMXTP("C",TPID,""))     Q IDER  GETPENDEI(CHFROM,CHTO) ; GET CHCLEI FROM PENDING ^CHMPAY(I) FROM/TO VALUES     K ^ZSC(\$J)     N IDX,CHJK,PDI,CHCLEI,CHCLAI,CTPI,RECCNT S RECCNT=0     F IDX=CHFROM:1 Q:IDX&gt;CHTO D     .S CHJK=\$O(^CHMPAY(IDX,"PDI",0)) Q:CHJK=""     .S PDI=\$P(^CHMPAY(IDX,"PDI",CHJK,0), "^",1) Q:PDI=""     .S CHCLEI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",6) Q:CHCLEI=""     .S CHCLAI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",3) Q:CHCLAI=""     .S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM </pre>	

Routine Name	CHMXWB21
	<pre> .I CHCLEI]" I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1 S ^ZSC(\$J,0)=RECCNT Q  <b>GETFINALEI(CHFROM,CHTO) ; GET CHCLEI FOR FINAL CLAIM STATUS</b> K ^ZSC(\$J),^ZSC1(\$J) N CHCLEI,CHCLAI,CTPI,RECCNT,CLC,CLB,CHPIDX,CHPDATE,CHPPDI S RECCNT=0 S %H=\$H-180 D YMD^%DTC S CHPDATE=X F S CHPDATE=\$O(^CHMPAY("E",CHPDATE)) <b>Q:CHPDATE="" S CHPIDX="" F S CHPIDX=\$O(^CHMPAY("E",CHPDATE,CHPIDX))</b> <b>Q:CHPIDX="" I \$G(^CHPIDX))'=1 D</b> . S CHPPDI=\$O(^CHMPAY(CHPIDX,"PDI",0)) Q:CHPPDI="" . I CHPPDI=\$P(\$G(^CHMPAY(CHPIDX,"PDI",PDI,0)),")^") . I CHPPDI="" Q . S CHCLEI=\$O(^CHMXCLE("D",CHPPDI,"")) I CHCLEI="" Q . I \$\$FINAL(CHCLEI) D .. S CLC=\$P(^CHMXCLE(CLE,0),")^") .. S CLB=\$P(^CHMXCLC(CLC,0),")^") .. S CHCLAI=\$P(^CHMXCLB(CLB,0),")^") .. S CTPI=\$\$GETTP(CHCLAI) ; <b>TRADING PARTNER FOR CURRENT CLAIM</b> .. I \$D(CHTPARR(CTPI)) <b>S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1,^ZSC1(\$J,CHPDATE,CHPIDX)=""</b> <b>S ^ZSC(\$J,0)=RECCNT</b> <b>Q</b>  ; ;..... ; GETCTPI RETURN THE NEXT TRADING PARTNER INDEX BASED ON THE LAST INDEX ; ; USED. THIS FUNCTRION IS CALLED BY THE "PENDING STATUS" PROCESS. ; ; FUNCTION USES LOCAL VARUIABLE "CHTPI" AS THE BASIS FOR RETURNING ALL ; ; TRADING PARTNER INDICES. IF TRADING PARTNER IS INACTIVE, FUNCTION MOVES ; ; TO THE NEXT TRACKING GLOBAL TRADING PARTNER. ; ;..... ;  GETCTPI N TRKI,CHTPI S CHTPI="" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .I (\$P(\$G(^CHMX277(TRKI,0)),")^",3)=1) S CHTPARR(CHTPI)=" Q  GETCLMI(,DATE,FLAG) ; MATCH A DATE WITH THE ^CHMXCL(I) ENTRY ; DATE DATE FOR THE ^CHMXCL() SEARCH ; FLAG "&gt;","&lt;","&lt;","&gt;" DATE SEARCH POSSIBILITIES N CHCNT,LSTI,EXIT,WDATE,RTN S RTN=0,EXIT=0 S LSTI=(\$P(^CHMXCL(0),")^",3))+1 F CHCNT=1:1 Q:(LSTI="")!(EXIT) S LSTI=\$O(^CHMXCL(LSTI),-1) D .S WDATE=\$P(\$P(\$G(^CHMXCL(LSTI,0)),")^",1),") ; FILE OPENED DATE .I ((FLAG="&lt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &gt; SPECIFIED DATE .E I ((FLAG="&lt;")&amp;(WDATE'&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE = OR &gt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE'&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; OR = SPECIFIED DATE Q RTN </pre>



Routine Name	CHMXWB21
	<pre> ..... ; FILEMAN UPDATE "DR" STRING BUILD FUNCTION ; .....  SETDARR(STR)   N REG,DRDATA,TMP,IDX,DARR   S DARR="",DRDATA=""   F IDX=1:1 S TMP=\$P(STR,"^",IDX) Q:TMP="" D   .S REG=\$P(TMP,"=",1),VAL=\$P(TMP,"=",2)   .S DARR(REG)=VAL   S DRDATA=\$\$SETDR^CHHRLIBFM("DARR")   Q DRDATA  ..... ; RETRIEVE VALUES FROM THE TRACKING GLOBAL ; ; NEEDS: CHTPI: TRADING PARTNER INDEX TO USE ; ; RUNTYPE: STATUS TYPE BEING CREATED ; .....  GETTOFROM(CHTPI) ; RETRIEVE TO-FROM FILES FROM TRACKING GLOBAL   N JDX,NODE,TRKI   S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; CONVERT CHTPI FOR TRACKING GLOBAL ACCESS   S NODE=\$S(RUNTYPE="P":20,RUNTYPE="F":30) ; SELECT PENDING/FINAL STATUS NODE   S JDX="A",JDX=\$O(^CHMX277(TRKI,NODE,JDX),-1) ; LOOK AT LAST ENTRY FOR THE STATUS   ;W !,"GETTOFROM: CHTPI = ",CHTPI," NODE= ",NODE," JDX= ",JDX   Q:JDX=""   S CHFROM=(\$P(\$G(^CHMX277(TRKI,NODE,JDX,0)),"^",4)) ; NEW "FROM"= OLD "TO"   S:CHFROM'=0 CHFROM=CHFROM+1,CHTO=\$P(\$G(^CHMPAY(0)),"^",3) ; SET START/ END INDEX   ;W !,"GETTOFROM: FROM= ",CHFROM," TO= ",CHTO   Q  ..... ; SUPPORT FUNCTIONS FOR THE FILE GENERATION FUNCTIONS DEFINED ABOVE. ; .....  DATERANGE(FROM,TO) ;Prompt for a FROM and TO date range.   N MAXTO   D NOW^%DTC S MAXTO=\$E(%,1,7)   S (FROM,TO)="", (DONE,POP)=0   F Q:(DONE)!(POP) D   .S FROM=\$\$GETDATE^CHMXWBUT("ENTER START DATE: ") S:FROM=-1 POP=1 Q:POP   .I ((FROM&lt;1)!(FROM&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S TO=\$\$GETDATE^CHMXWBUT("ENTER END DATE: ") S:TO=-1 POP=1 Q:POP   .I ((TO&lt;FROM)!(TO&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S DONE=1   Q POP  GETSTVAL(CHCLEI) ; REVERSE POLISH METHOD FOR RETRIEVING CLAIM STATUS   N CHCLCI,CHCLBI,CHCLAI,CHCLI,TARGEI,STAT,PDI   S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES </pre>

Routine Name	CHMXWB21
	<pre> S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S PDI=\$P(^CHMXCLE(CHCLEI,100),"^",2) S:PDI="" PDI=\$P(^CHMXCLE(CHCLEI,100),"^",4) I PDI="" Q 6 ;Assume 6 that it was rejected if no PDI found...data integrity issue S STAT=0,TARGEI=0 F Q:TARGEI=CHCLEI D .S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) .S CHIDXS=0,CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,"")) .S TARGEI=\$P(CHIDXS,"*",3) Q STAT  ***** ; EMAIL STATUS TO INTERESTED PARTIES ; *****  AUTOMM(TYPE) D NOW^%DTC S FMDATE=\$E(%,1,7) S CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL_ (Total Records = "_CHBRCCNT_") S CHNB=CHNB+1,ZML(CHNB)=" " I TYPE="EFILE" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: ***ERROR*** CREATING 277U STAT file!" E I TYPE="OK" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: Successful CREATION 277U STAT file!" S CHNB=CHNB+1,ZML(CHNB)="EDI BATCH NUMBER ^CHMXCL("_CHMXI_",0)" S XMTEXT="ZML(",XMSUB="277U WebMD..Success.."_FMDATE S XMDUZ=.5 S XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="" S XMY("FRANK.PELUSO@VA.GOV")="" D ^XMD Q  ***** ; THE FOLLOWING ROUTINES ARE PRESERVED FOR HISTORICAL PURPOSES ; ; THERE ARE COPIES OF THESE FUNCTIONS IN CHMXWB07,CHMXWB11,CHMXWB21 ; *****  ; EXFTP S CHFTPMSG=0 ;H 1000 ;DEF4399 ajm removed hang, not needed XCOM S X=\$ZF(-1,"SUBMIT DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP.COM /NAME=WEB_WS_FTP_JOB/NOPRINTER/USER=DSMMANAG/PARAM="_CHFILE) ;//////// UNREMARK/REMARK OUT CODE FOR PRODUCTION ////////// ;H 1200 ;DEF4399 ajm removed hang, not needed K ^CHMZHOLD("RKN_W277") Q ;DEF4399 ajm addd - files are copied, no longer ftp'd ; FTPCK S RD="DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP_JOB.LOG" ;LIVE C RD K CHFTPTIM,CHFTPWT S ZE="",QFLAG=0,CHFTPWT=\$P(^CHMDIC(741002.17,1,2),"^",10) F CHFTPTIM=1:1:CHFTPWT H 60 O RD:"R" D Q:(QFLAG=1) ;RKN, 6-29-05, MOD READONLY .I \$T C RD Q </pre>

Routine Name	CHMXWB21
<pre> .F U RD R RDLIN D Q:(QFLAG=1)!(\$ZE["ENDOFIL"]) ..I (RDLIN["226 Transfer complete"]) S CHFTPMSG=1,QFLAG=1 Q ..I (RDLIN["Charged CPU time:"]) S QFLAG=1 Q FTPERR C RD S CHNB=2 S ZML(CHNB)=" " S ZML(CHNB)=CHVMSFL S CHNB=CHNB+1 S ZML(CHNB)=" " S CHNB=CHNB+1 S XMTEXT="ZML(" K CHFTPMSG,RD,RDLIN ; ; ; ; CALCULATE A DATE 90 DAYS IN THE PAST ; ; ; DOLDFLS D NOW^%DTC S X1=X,X2=-90 D C^%DTC D YX^%DTC Q ;  ; ; Error Handler / personnel notification functions for the 277 status file ; ; generation process. ; ;  VMSERR C IO D ^%ZTER ; ERRMM ; ;S CHERRMSG=1,CHNB=2,ZML(CHNB)=" ",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)=" ",CHNB=CHNB+1 ;S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 ;S ZML(CHNB)=" 277U WebMD file!" ;S XMDUZ=.5,XMY("10722")=" " ;RKN LIVE ;S XMTEXT="ZML(",XMSUB="277U WebMD..Error.."_FMDATE S CHERRMSG=1,CHNB=2,ZML(CHNB)=" ",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)= ",CHNB=CHNB+1 S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 S ZML(CHNB)=" CSTAT file!" ;S XMDUZ=.5,XMY("JAMES.NELSON5@VA.GOV")=" " ;RKN LIVE ;S XMY("SREEDHAR.MADANAPALLE@MED.VA.GOV")=" " S XMDUZ=.5,XMY("274577")=" ",XMY("246183")=" " S XMY("AHMED.MAGDUB@VA.GOV")=" ",XMY("FRANK.PELUSO@VA.GOV")=" " S XMTEXT="ZML(",XMSUB="277U..Error.."_DATESTAMP ; BAS -END MOD D ^XMD Q </pre>	

Routine Name	CHMXWB21
<pre> ..... ; \$TEXT VARIABLE SETUP ROUTINES ; ..... </pre>	
<pre> GETHDRVAL   S LOADTYPE=\$S(RUNTYPE="H": "F", 1: "I")   S (PAYNAME, PAYPHONE)= ""   I RUNTYPE="F" S PAYNAME="CHAMPVA CUSTOMER SERVICE", PAYPHONE="8007338387"   Q </pre>	
<pre> GETCLMVAL   N CHCLB0, CHCLB1, CHMPAY1, CHCLE60   N TMPDTE S TMPDTE=0   S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))   S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))   S CHCLE64=\$G(^CHMXCLE(CHCLEI,64))   S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))   S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID   S BPPN=""   ; 6.BILLING PROVIDER PAYER NUMBER   S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID   S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME   S BPFNAME="" ;   9.BILLING PROVIDER FIRST NAME   S BPMNAME=""   ; 10.BILLING PROVIDER MIDDLE NAME   S BPNAMEx="" ;   11.BILLING PROVIDER NAME SUFFIX   S SPFTID=""   ; 12.RENDERING PHYSICIAN FEDERAL TAX ID   S SPPN=""   ; 13.RENDERING PHYSICIAN PAYER NUMBER   S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID   S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME   S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME   S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME   S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX   S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER   S EMPNAME="" ; 20.EMPLOYER NAME   S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))   S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID   S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME   S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME   S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME   S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX   S PATID="" ; 26.PATIENT ID   S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME   S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME   S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME   S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX   S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH   S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER   S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN   S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER </pre>	

Routine Name	CHMXWB21
S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)>0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1)) ; 34.CHARGE AMT <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b> <b>AMOUNT</b> <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b> <b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$S(RUNTYPE="F":",",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),"^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER S VOUCHID="" ; 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47. PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1 INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2 DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q  GPSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S STDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11),TMPDTE=STDATE ; TMPDTE IS LAST START DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:CHFI D ; CHECK ALL LI "FROM" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11) .I TDATE<STDATE S STDATE=TDATE ; TRACK "OLDEST" START DATE .I TDATE>TMPDTE S TMPDTE=TDATE ; TRACK "MOST RECENT" START DATE S EDATE=\$\$GVCEND S:(EDATE="")!(EDATE<TMPDTE) EDATE=TMPDTE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES  GISTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) ; END DATE S SVCDATES=STDATE_"*"_EDATE	

Routine Name	CHMXWB21
<p>Q SVCDATES</p> <p>GDSTARTEND() ; POPULATE EMDEON FIELD FOR DENTAL CLAIM START/END DATES  N STDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>GETSTCVAL(TYPE)  S PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 4.PAYER CLAIM ID NUMBER(PDI)  S:PCINUM="" PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED,  DERIVED CLAIM CTL NUM  S DATERR="" ; 10.EMDEON SPECIFIC DATA IN ERROR  S ESCODE="" ; 11.EMDEON SPECIFIC STATUS CODE  <b>I RUNTYPE="F" D Q</b>  <b>. I \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICNUM=\$P(^",^",23)</b>  <b>. I \$G(LICNUM)= "" S LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2)</b>  S:TYPE="CLM" LICNUM="" ; 5.CLM LVL LINE ITEM CONTROL NUMBER  S:TYPE="DTL" LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; SERVICE LINE NUMBER  Q</p> <p>.....  ; 9/7/2011: modified Procedure Modifier #1 extract to field 4 from field 3;  .....</p> <p>GDTLVAR\$ ; DTL RECORD DATA GATHERING  S LICN=""  S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER  S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED  CLAIM CTL NUM  S CHCLF1=\$G(^CHMXCLF(CHCLFI,1))  <b>I RUNTYPE="F",\$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICN=\$P(^",^",23)</b>  I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER  S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID  S:SQID="" SQID="NU" ;  DEFAULT VALUE IF NOT PROVIDED  S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE  S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT  PROVIDED  S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1  S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2  S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3  S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4  S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT  S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL  LINE ITEM CHARGE AMOUNT  S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND  LINE ITEM PROV. PAYMENT AMOUNT  S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE  ITEM PROV. PAYMENT AMOUNT  S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE  S:RVNUCODE="9999" RVNUCODE="" ; DON'T  OUTPUT 9999 VALUE  S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF</p>	

Routine Name	CHMXWB21
<p>SERVICE)</p> <p>S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL</p> <p>QUANTITY(UNITS OF SERVICE)</p> <p>S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 16.EMDEON CLAIM NUMBER</p> <p>S CLMTYP=\$\$GETCLMTYP ; GET</p> <p>I/P/D CLAIM TYPE</p> <p>S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17.</p> <p>PROFESSIONAL SERVICE START/END DATE</p> <p>S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1</p> <p>INSTITUTIONAL SERVICE START/END DATES</p> <p>S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL</p> <p>SERVICE START/END DATES</p> <p>S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2)</p> <p>Q</p> <p>.....</p> <p>; LINE ITEM FUNCTIONS TO GATHER THE SPECIFIC DATA FIELDS ;</p> <p>.....</p> <p>.....</p> <p>; EMDEON REQUIRES THE EARLIEST START DATE (CLAIM OR LINE ITEM DATE ;</p> <p>; FOR THE SERVICE START DATE. THIS FUNCTION 1) GETS THE CLAIM START ;</p> <p>; DATE, THEN SEARCHES THROUGH LINE ITEMS FOR ANY EARLIER DATES. ;</p> <p>; THE EARLIEST DATE FOUND IS RECORDED AS THE START DATE. ;</p> <p>.....</p> <p>.....</p> <p>; DETERMINE THE CLAIM TYPE, I.E. PROFESSIONAL, INSTITUTIONAL, OR DENTAL</p> <p>; 4010/5010 VERSION IDENTIFIERS</p> <p>; 98 / 222 PROFESSIONAL CLAIMS</p> <p>; 96 / 223 INSTITUTIONAL CLAIMS</p> <p>; 97 / 224 DENTAL CLAIMS</p> <p>.....</p> <p>.....</p> <p>GETCLMTYP()</p> <p>N TYPE ; VERSION-FUNCTIONAL TYPE NUMBER FROM ^CHMXCLA()</p> <p>S TYPE=\$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13)</p> <p>Q:(TYPE[222]!(TYPE[98] "P"</p> <p>Q:(TYPE[223]!(TYPE[96] "I"</p> <p>Q:(TYPE[224]!(TYPE[97] "D"</p> <p>Q "U"</p> <p>.....</p> <p>; PROFESSIONAL START/END DATES</p> <p>.....</p> <p>GPDSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL START/END DATES</p> <p>N STDATE,CHFI,TDATE,EDATE,SVCDATES</p> <p>S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE</p> <p>S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE</p> <p>S:EDATE="" EDATE=STDATE</p> <p>S SVCDATES=STDATE_ "*" _EDATE</p> <p>Q SVCDATES</p>	

Routine Name	CHMXWB21
<p>GIDSTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL DETAIL START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11)  S:STDATE="" STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12)  S:EDATE="" EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11)  S:EDATE="" EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2)  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>GDDSTARTEND() ; EMDEON DENTAL DETAIL START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) ; START DATE  S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) ; END DATE  S:EDATE="" EDATE=STDATE  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>.....  ; EMDEON REQUIRES THE LAST END DATE (CLAIM OR LINE ITEM DATE ;  ; FOR THE SERVICE END DATE. THIS FUNCTION 1) GETS THE CLAIM START ;  ; DATE, THEN 2)SEARCHES THROUGH LINE ITEMS FOR ANY LATER DATES. ;  ; THE LATEST DATE FOUND IS RECORDED AS THE END DATE. ;  .....</p> <p>GSVCEND() ; POPULATE EMDEON REQUIRED FIELD FOR SERVICE END DATE  N ENDATE,CHFI,TDATE  S CHFI=0,CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI))  S ENDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12) ; ^CHMXCLF(I) END DATE  F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:CHFI D ; CHECK ALL LI "TO" DATES  .I TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12)  .I TDATE&gt;ENDATE S ENDATE=TDATE ; TRACK "YOUNGEST" END  DATE  Q ENDATE</p> <p>GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT  Q 0</p> <p>GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT  Q 0</p> <p>GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE)  Q 0</p> <p>.....  ; DOCUMENTREC (UTILITY THAT HELPS IN DOCUMENTING THE RECORD GENERATION  PROCESS) ;  ; CREATES A FILE THAT CONTAINS THE RECORD INFORMATION FOR THE 5010 ;  ; EMDEON STATUS RECORDS, INCLUDING THE HEADER, CLAIM, LINE ITEM, AND  TRAILER ;  ; RECORDS. THIS FUNCTION USES THE FIELD DESCRIPTORS TO DOCUMENT EACH  FIELD IN ;  ; THE RECORDS, I.E.: ;  ; 1) RECORD NAME ;  ; 2) STARTING LOCATION IN THE RECORD ;</p>	



Routine Name	CHMXWB21
<pre> ; 3) LENGTH (WIDTH) OF THE FIELD ; ; 4) JUSTIFICATION WITHIN THE FIELD ; ; 5) THE VALUE (HARD CODED FIELDS) OR THE CACHE FILELOCATION FROM WHICH THE ; ; VALUE IS RETRIEVED. ; ..... </pre>	
<p>DOCUMENTREC</p> <pre> N DOCFILE,TMPIO,LN,HTABS,FTABS,RTYPE,RNAME,CHTYPE,STR,COLNMS S RTYPE="EMDEONHDR^EMDEONCLM^EMDEONSTC^EMDEONTRLR^EMDEONLI" ; NAMES OF \$TEXT TABLES S COLNMS="FIELD NAME^USE^DESC^PAD^JUST^VALUE" ; COLUMN HEADER </pre>	
<p>NAMES</p> <pre> S HTABS="35^39^45^50^57" ; HEADER TAB STOPS FOR THE FIELD </pre>	
<p>DESCRIPTIONS</p> <pre> S FTABS="36^39^46^52^57" ; FIELD TAB STOPS FOR DESCRIPTIONS S DOCPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" S DOCFILE="DOC277_5010_"_DATESTAMP_.TXT" ; STATUS MAPPING </pre>	
<p>DOCUMENTATION</p> <pre> S DOCFILE=DOCPATH_DOCFILE ; OUTPUT THE \$TEXT </pre>	
<p>TO A PRINTABLE FILE</p> <pre> S FLAG=\$\$OFIL^CHMXWBUT(DOCFILE,"NWS") S TMPIO=\$IO U DOCFILE F CHTYPE=1:1 S RNAME=\$P(RTYPE,"^",CHTYPE) Q:RNAME="" D .W !!,?20,"HEALTH CARE CLEARING HOUSE """,RNAME,"" RECORD DEFINITIONS" ; TITLE .W !,\$P(COLNMS,"^",1),\$P(HTABS,"^",1),\$P(COLNMS,"^",2),\$P(HTABS,"^",2),\$P(COLNMS,"^",3), \$P(HTABS,"^",3),\$P(COLNMS,"^",4),\$P(HTABS,"^",4),\$P(COLNMS,"^",5),\$P(HTABS,"^",5),\$P(COLNMS,"^",6), \$P(HTABS,"^",6),\$P(COLNMS,"^",7) .F LN=1:1 S STR=\$T(@RNAME+LN) Q:STR["END OF RECORD" D ; READ \$TEXT </pre>	
<p>DESCRIPTOR</p> <pre> ..W !,\$P(STR,";",3),\$P(FTABS,"^",1),\$P(STR,";",10),\$P(FTABS,"^",2),\$P(STR,";",8),\$P(FTABS,"^",3), \$P(STR,";",7),\$P(FTABS,"^",4),\$P(STR,";",6),\$P(FTABS,"^",5),\$P(STR,";",4) U TMPIO D CLOSEFILE^CHMXWBUT(DOCFILE) ; CLOSE CURRENT FILE Q </pre>	
<pre> ..... ; EMDEONHDR: Common Header for EMDEON STATUS Files ; ; A single header is generated for each output file. ; ; DESC: "FIELD NAME";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA TYPE"; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR(record # and text description) ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT, C= CENTER ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH (ANY PRINTABLE CHARACTER) ; ; NOTE: PAD CHAR="" IF NO CHARACTER IS BETWEEN THE SEMICOLONS (I.E. ;;) ; ; NO PADDING WILL OCCUR IF THIS IS SET UP THIS WAY ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; ; ; FORMATDATA TREATS THE PAD CHAR (:) AS A NULL, SO NO PADDING OCCURS. ; ; THIS WILL ALLOW USE OF THE FORMATDATA FUNCTION WITHOUT MODIFICATION </pre>	

Routine Name	CHMXWB21
<p>BETWEEN ;</p> <p>    ; PADDED AND NON-PADDED FIELDS. ;</p> <p>.....</p> <p>    ; 8/4/11 DLB "6. CREATION TIME" FROM \$(DATESTAMP,9,12) TO \$(DATESTAMP,9,14) ;</p> <p>    ; 8/15/11 DLB "2. FILE GROUP ID" INSERTED THE DATESTAMP VALUE TO ENSURE</p> <p>UNIQUENESS ;</p> <p>    ; 9/7/2011 DLB 12. LOAD TYPE CHANGED TO PROVIDE "F" WHEN HISTORICAL FILE</p> <p>GENERATED ;</p> <p>.....</p> <p>EMDEONHDR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";DATA</p> <p>LENGTH/PATTERN";FIELD START LOCATION;FIELD USE</p> <p>    ;;1.RECORD ID;"HDR";3;L;;3AN;0;R;</p> <p>    ;;2.FILE GROUP ID;\$(GROUPID="'":GROUPID,1:DATESTAMP);20;L;;20AN;4;R;</p> <p>    ;;3.FILE GROUP SEQUENCE NUMBER;FILECNT;2;R;0;3N;24;R;</p> <p>    ;;4.FILE GROUP COUNT;GRPCNT;2;R;0;3N;26;R;</p> <p>    ;;5.CREATION DATE;\$(DATESTAMP,1,8);8;L;;8AN;28;R;</p> <p>    ;;6.CREATION TIME;\$(DATESTAMP,9,14);6;L;;6N;36;R;</p> <p>    ;;7.TRADING PARTNER ID;"VAFNH";10;L;;10AN;42;R;</p> <p>    ;;8.SUBMITTER NAME;"VA HEALTH ADMIN CENTER";30;L;;30AN;53;R;</p> <p>    ;;9.PAYER CONTACT NAME;PAYNAME;60;L;;60AN;83;O;</p> <p>    ;;10.PAYER SUPPORT TELEPHONE NUMBER;PAYPHONE;10;L;;10N;143;O;</p> <p>    ;;11.PAYER SUPPORT EMAIL ADDRESS;"";80;L;;80AN;153;O;</p> <p>    ;;12.LOAD TYPE;LOADTYPE;1;L;;1AN;233;R;</p> <p>    ;;13.PAYER UNIQUE FILE IDENTIFIER;DATESTAMP;20;L;;20AN;234;R;</p> <p>    ;;14.FILE TYPE;"CStat";5;L;;5AN;254;R;</p> <p>    ;;15.VERSION CODE;"03";2;L;;2AN;258;R;</p> <p>    ;;16.RELEASE CODE;"00";2;L;;2AN;260;R;</p> <p>    ;;18.END OF RECORD;</p> <p>.....</p> <p>    ; EMDEON CLAIM IEVEL STATUS TABLE: Contains the Descriptions for all data to be ;</p> <p>    ; gathered for E01 File. Used by \$TEXT to gather all data &amp; set up format ;</p> <p>    ; Each record is described in detail for the following Parameters: ;</p> <p>    ; This table is used as a block of data(a record) for each claim to be processed. ;</p> <p>    ; Therefore, separating this from the header and trailer tables allows multiple ;</p> <p>    ; calls to this table. ;</p> <p>.....</p> <p>    ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ;</p> <p>    ; FIELD NAME: EMDEON File FIELD DESCRIPTOR ;</p> <p>    ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ;</p> <p>    ; JUSTIFY FLAG: L=LEFT, R=RIGHT ;</p> <p>    ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ;</p> <p>    ; PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ;</p> <p>    ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION</p> <p>ONLY ;</p> <p>    ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ;</p> <p>.....</p> <p>    ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP) ;</p> <p>    ; 8/5/11 DLB FIELDS 12,14,15: TYPO FIXED FOR PROVIDER INFO (ADDED ^ TO CHMXCLB) ;</p> <p>    ; 8/5/11 DLB FIELD 41: CHANGED INDEX FROM ^CHMXCLE(I,100 TO ^CHMXCLE(I,0 ;</p> <p>.....</p> <p>EMDEONCLM ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA</p> <p>LENGTH/PATTERN";FIELD START LOCATION;FIELD USE</p>	

Routine Name	CHMXWB21
	<pre> ;;1.RECORD ID;"CLM";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.MAINTENANCE TYPE CODE;\$\$(\$G(RUNTYPE)="F":"021",\$G(RUNTYPE)="A":"021",\$G(RUNTYPE)="P":"021",\$G(RUNTY PE)="H":"030",1:"001");3;L;;3AN;19;R; ;;5.BILLING PROVIDER FEDERAL TAX ID;BPFTI;9;L;;9N;22;C; ;;6.BILLING PROVIDER PAYER NUMBER;BPPN;50;L;;50AN;31;C; ;;7.BILLING PROVIDER NATIONAL ID;BPNID;10;L;;10N;81;C; ;;8.BILLING PROVIDER LAST NAME;BPLNAME;60;L;;60AN;91;C; ;;9.BILLING PROVIDER FIRST NAME;BPFNAME;35;L;;35AN;151;O; ;;10.BILLING PROVIDER MIDDLE NAME;BPMNAME;25;L;;25AN;186;O; ;;11.BILLING PROVIDER NAME SUFFIX;BPNAMEX;10;L;;10AN;211;O; ;;12.SERVICE PROVIDER FEDERAL TAX ID;SPFTID;9;L;;9N;221;C; ;;13.SERVICE PROVIDER PAYER NUMBER;SPPN;50;L;;50AN;230;C; ;;14.SERVICE PROVIDER NATIONAL ID;SPNID;10;L;;10N;280;C; ;;15.SERVICE PROVIDER LAST NAME;SPLNAME;60;L;;60AN;290;C; ;;16.SERVICE PROVIDER FIRST NAME;SPFNAME;35;L;;35AN;350;O; ;;17.SERVICE PROVIDER MIDDLE NAME;SPMNAME;25;L;;25AN;385;O; ;;18.SERVICE PROVIDER NAME SUFFIX;SPNAMEX;10;L;;10AN;410;O; ;;19.EMPLOYER IDENTIFICATION NUMBER;EMPIDNUM;80;L;;80AN;420;C; ;;20.EMPLOYER NAME;EMPNAME;60;L;;60AN;500;C; ;;21.SUBSCRIBER ID;SUBSCID;80;L;;80AN;580;C; ;;22.SUBSCRIBER LAST NAME;SUBLNAME;60;L;;60AN;660;C; ;;23.SUBSCRIBER FIRST NAME;SUBFNAME;35;L;;35AN;720;O; ;;24.SUBSCRIBER MIDDLE NAME;SUBMNAME;25;L;;25AN;755;O; ;;25.SUBSCRIBER NAME SUFFIX;SUBNAMEX;10;L;;10AN;780;O; ;;26.PATIENT ID;PATID;80;L;;80AN;790;O; ;;27.PATIENT LAST NAME;PATLNAME;60;L;;60AN;870;R; ;;28.PATIENT FIRST NAME;PATFNAME;35;L;;35AN;930;R; ;;29.PATIENT MIDDLE NAME;PATMNAME;25;L;;25AN;965;O; ;;30.PATIENT NAME SUFFIX;PATNAMEX;10;L;;10AN;990;O; ;;31.PATIENT DATE OF BIRTH;PATDOB;8;L;;8DT;1000;R; ;;32.PATIENT GENDER;PATGENDR;1;L;;1AN;1008;O; ;;33.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;1009;O; ;;34.CLAIM CHARGE AMOUNT;CLMCHRG;18;L;;18N;1059;R; ;;35.CLAIM PYMT AMT;CLMPMT;18;R;;18N;1077;C; ;;36.CLAIM ADJ/PAYMENT DATE;CAPD;8;L;;8DT;1095;C; ;;37.CHECK/EFT DATE;CHKEFTDATE;8;L;;8DT;1103;O; ;;38.CHECK/EFT NUMBER;CHKEFTNUM;16;L;;16AN;1111;O; ;;39.BILL TYPE;BTYPE;3;L;;3AN;1127;O; ;;40.PAYER CLAIM ID NUMBER;PCIDNUM;50;L;;50AN;1130;R; ;;41.PATIENT ACCOUNT NUMBER;PATACT;50;L;;50AN;1180;O; ;;42.PHARMACY PRESCRIPTION NUMBER;PRSCNUM;50;L;;50AN;1230;O; ;;43.VOUCHER IDENTIFIER;VOUCHID;50;L;;50AN;1280;O; ;;44.APP/LOCATION SYSTEM ID;LOCSYSID;50;L;;50AN;1330;O; ;;45.GROUP NUMBER;GRPNUM;50;L;;50AN;1380;O; ;;46.CLAIM SERVICE DATE START;CLMSTDT;8;L;;8DT;1430;R; ;;47.CLAIM SERVICE DATE END;CLMENDT;8;L;;8DT;1438;R; ;;48.END OF RECORD;  ..... ; EMDEON CLAIM STATUS RECORD DETAIL: Contains the Descriptions for all data to be ; ; gathered for Claim level and Line Level records. Used by \$TEXT to gather all ; ; data &amp; set up format. ; </pre>

Routine Name	CHMXWB21
	<pre> ; Each record is described in detail for the following Parameters: ; ; NOTE: ; ; For each CLM record, there must be at least one STC record. ; ; For a single CLM record, The following is true about DTL records: ; ; 1) Payer can send any number of DTL records (including NONE) ; ; 3) Each DTL record sent corresponds to Line Level info for the CLM record. ; ; 4) For each DTL record sent, there must be at least one STC record. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----; ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP) ; ;-----;  EMDEONSTC ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"STC";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;3;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;13;R; ;;4.PAYER CLAIM ID NUMBER;PCINUM;50;L;;50AN;18;R; ;;5.LINE ITEM CONTROL NUMBER;LICNUM;50;L;;50AN;68;O; ;;6.STATUS INFORMATION EFF. DATE;DATESTAMP;14;L;;14N;118;R; ;;7.CLAIM STATUS CATEGORY CODE;RJCODE;3;L;;3AN;126;R; ;;8.CLAIM STATUS CODE;RJSTATUS;3;L;;3AN;129;R; ;;9.ENTITY CODE;ENTITY;3;L;;3AN;132;O; ;;10.DATA IN ERROR;DATERR;264;L;;264AN;135;O; ;;11.EMDEON STATUS CODE;ESCODE;5;L;;5AN;399;O; ;;13.END OF RECORD;  ;-----; ; EMDEONTRLR: Common Trailer for EMDEON Status Files ; ; NOTE: a single trailer record is generated for each output file. ; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: 277 File FIELD DESCRIPTOR ; ; LENGTH: 277 FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----;  EMDEONTRLR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"TRLR";4;L;;4AN;0;R; ;;2.RECORD COUNT;COUNT;10;R;;10N;4;R; ;;3.END OF RECORD; </pre>

Routine Name	CHMXWB21
<pre> ..... ; EMDEON LINE LEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for CRF File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; NOTE: there may be multiple line items for each claim record. This table ; ; defines the record to be generated for each line item. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; .....  EMDEONLI ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"DTL";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.PAYER CLAIM ID NUMBER;PCID;50;L;;50AN;19;R; ;;5.LINE ITEM CONTROL NUMBER;LICN;50;L;;50AN;69;R; ;;6.SERVICE QUALIFIER ID;SQID;2;L;;2AN;119;R; ;;7.SERVICE IDENTIFICATION CODE;SICODE;48;L;;48AN;121;R; ;;8.PROCEDURE MODIFIER 1;PRCM1;2;L;;2AN;131;O; ;;9.PROCEDURE MODIFIER 2;PRCM2;2;L;;2AN;133;O; ;;10.PROCEDURE MODIFIER 3;PRCM3;2;L;;2AN;135;O; ;;11.PROCEDURE MODIFIER 4;PRCM4;2;L;;2AN;137;O; ;;12.LI CHARGE AMOUNT;LICHARGA;18;L;;18N;139;R; ;;13.LINE ITEM PROV. PAYMENT AMOUNT;LIPPA;18;L;;18N;157;R; ;;14.REVENUE CODE;RVNUCODE;48;L;;48AN;175;C; ;;15.QUANTITY(UNITS OF SERVICE);QTYUOS;15;L;;15N;223;O; ;;16.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;238;O; ;;17.SRVC START DATE;SVCSTDAT;8;L;;8N;288;R; ;;18.SRVC END DATE;SVCENDAT;8;L;;8N;306;R; ;;20.END OF RECORD; </pre>	

### 3.2.1.5 Add Line Item Control Number (LICN) to the Claim Line Level Detail Record for Primary Claims

#### Functional Requirement: 2.8.1.5 LICN – Claim Line Level Record Details – Primary Claims

The CP&E system shall transmit to the HCCH, in the Final Claim Status file, the Line Item Control Number in the Claim Line Level Record Details based on the following logic:

- When available - Line Item Control Number with Qualifier = 6R (Provider Control Number)

- Else - Assigned Number

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.1.5	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		
<b>LINE ; BUILD DTL RECORDS FOR LINE ITEMS</b> N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D <b>.D GDTLVARS</b> .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),",") .. I HACCLM="" D ... S CHRJARR(0)=1		

Routine Name	CHMXWB21
<pre> ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65*" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1*" Q ... S CHRJARR(1,1)="F0*1*" ... Q .. Q .D BLDSTC("DTL") Q </pre>	
GDTLVAR	; DTL RECORD DATA GATHERING
S LICN=""	
S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER	
S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED	
CLAIM CTL NUM	
S CHCLF1=\$G(^CHMXCLF(CHCLFI,1))	
I RUNTYPE="F",\$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICN=\$P(^",^",23)	
I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER	
S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID	
S:SQID="" SQID="NU"	;
DEFAULT VALUE IF NOT PROVIDED	
S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE	
S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT	
PROVIDED	
S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1	
S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2	
S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3	
S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4	
S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT	
S:RUNTYPE="F" LICHARGA=\$\$GETFLICA	; FINAL
LINE ITEM CHARGE AMOUNT	
S:RUNTYPE="F" LIPPA="0.00"	; 13.ACK/PEND
LINE ITEM PROV. PAYMENT AMOUNT	
S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA	; FINAL LINE
ITEM PROV. PAYMENT AMOUNT	
S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE	
S:RVNUCODE="9999" RVNUCODE=""	; DON'T
OUTPUT 9999 VALUE	
S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF	
SERVICE)	
S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS	; FINAL
QUANTITY(UNITS OF SERVICE)	
S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),^",5) ; 16.EMDEON CLAIM NUMBER	
S CLMTYP=\$\$GETCLMTYP	; GET
I/P/D CLAIM TYPE	
S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND	; 17.
PROFESSIONAL SERVICE START/END DATE	
S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND	; 17.1
INSTITUTIONAL SERVICE START/END DATES	
S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND	; 17.2 DENTAL
SERVICE START/END DATES	
S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2)	
Q	

### 3.2.1.6 Add LICN to the Claim Status Record Detail Record for Primary Claims

#### Functional Requirement: 2.8.1.6 LICN – Claim Status Record Details – Primary Claims

The CP&E system shall transmit to the HCCH, in the Final Claim Status file, the Line Item Control Number in the Claim Status Record Details based on the following logic:

- When available - Line Item Control Number with Qualifier = 6R (Provider Control Number)
- Else - Assigned Number

Routine Name	<b>CHMXWB21</b>		
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change		
Requirement Traceability Matrix	2.8.1.6		
Related Options			
Related Routines	Routines "Called By"	Routines "Called"	
	Taskman process	CHMXWBUT	
Data Dictionary (DD) References			
Related Protocols			
Related Integration Control Registrations (ICRs)			
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local		
Input Attribute Name and Definition	Name: Definition:		
Output Attribute Name and Definition	Name: Definition:		
Current Logic			
Same as 3.2.1.1			
Modified Logic (Changes are in bold and highlighted)	<p><b>LINE ; BUILD DTL RECORDS FOR LINE ITEMS</b></p> <p>N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D <b>.D GDTLVAR</b> .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "\$\$_\$FORMATDATA^CHMXWBUT(STR)</p>		



Routine Name	CHMXWB21
<pre> .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)), "^") .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)), "^",2)=4 S CHRJARR(1,1)="F1*65*" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)), "^",2)=0 S CHRJARR(1,1)="F2*1*" Q ... S CHRJARR(1,1)="F0*1*" ... Q .. Q .D BLDSTC("DTL") Q </pre>	
<pre> GDTLVARS ; DTL RECORD DATA GATHERING S LICN="" S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S CHCLF1=\$G(^CHMXCLF(CHCLFI,1)) I RUNTYPE="F", \$P(\$G(^CHMXCLF(CHCLFI,1)), "^",23)="" S LICN=\$P(^(" ",23) I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)), "^",2) ; 5.SERVICE LINE NUMBER S SQID=\$P(CHCLF1, "^",2) ; 6.SERVICE QUALIFIER ID S:SQID="" SQID="NU" ; DEFAULT VALUE IF NOT PROVIDED S SICODE=\$P(CHCLF1, "^",3) ; 7.SERVICE IDENTIFICATION CODE S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1), "^",1) ; DEFAULT VALUE IF NOT PROVIDED S PRCM1=\$P(CHCLF1, "^",4) ; 8.PROCEDURE MODIFIER 1 S PRCM2=\$P(CHCLF1, "^",5) ; 9.PROCEDURE MODIFIER 2 S PRCM3=\$P(CHCLF1, "^",14) ; 10.PROCEDURE MODIFIER 3 S PRCM4=\$P(CHCLF1, "^",15) ; 11.PROCEDURE MODIFIER 4 S:RUNTYPE="F" LICHARGA=\$P(CHCLF1, "^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND LINE ITEM PROV. PAYMENT AMOUNT S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1), "^",1) ; 14.REVENUE CODE S:RVNUCODE="9999" RVNUCODE="" ; DON'T OUTPUT 9999 VALUE S:RUNTYPE="F" QTYUOS=\$P(CHCLF1, "^",8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE) S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3), "^",5) ; 16.EMDEON CLAIM NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17. PROFESSIONAL SERVICE START/END DATE S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1 INSTITUTIONAL SERVICE START/END DATES </pre>	

Routine Name	<b>CHMXWB21</b>
S:CLMTYP="D" SVCDATES=\$GDDSTARTEND ; 17.2 DENTAL SERVICE START/END DATES S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2) Q	

### 3.2.1.7 Check/EFT Date for Split Claim Submission for Primary Claims

#### Functional Requirement: 2.8.1.7 Check/EFT Date for Split Submission – Primary Claims

The CP&E system shall use the most recent Check/EFT Date in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system and at least one of the claims contains a Check/EFT Date.

Routine Name	<b>CHMXWB21</b>		
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change		
Requirement Traceability Matrix	2.8.1.7		
Related Options			
Related Routines	Routines "Called By"	Routines "Called"	
	Taskman process	CHMXWBUT	
Data Dictionary (DD) References			
Related Protocols			
Related Integration Control Registrations (ICRs)			
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local		
Input Attribute Name and Definition	Name: Definition:		
Output Attribute Name and Definition	Name: Definition:		
Current Logic			
Same as 3.2.1.1			
Modified Logic (Changes are in bold and highlighted)	<b>BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS</b> <b>; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE</b> <b>; STVAL STATUS VALUE AFTER FROMNT END EDITS</b> <b>N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE</b> <b>I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE</b> <b>THIS</b>		

Routine Name	CHMXWB21
	<pre> .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+170000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ; TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ; GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D ..I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" ..I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,'D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^", 16) ..I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0,"^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK" </pre>

Routine Name	CHMXWB21
<b>FINAL(CHCLEI) ; FINAL STATUS</b> N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D . S CHMPAY=0,OK=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=0 Q .. S OK=0 .. Q Q OK	
<b>CLM ;</b> N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, <b>SUBLNAME</b> N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN <b>AMEX,PATDOB</b> N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P <b>CIDNUM</b> N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q	

### 3.2.1.8 Check/EFT Number for Split Claim Submission for Primary Claims

#### Functional Requirement: 2.8.1.8 Check/EFT Number for Split Submission – Primary Claims

The CP&E system shall use the most recent Check/EFT Number in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system and at least one of the claims contains a Check/EFT Number.

Routine Name	CHMXWB21
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change
Requirement Traceability Matrix	2.8.1.8
Related Options	

Routine Name	<b>CHMXWB21</b>	
<b>Related Routines</b>	<b>Routines "Called By"</b>	<b>Routines "Called"</b>
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)	<pre> <b>BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS</b>       ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE       ; STVAL STATUS VALUE AFTER FROMNT END EDITS       N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE       I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE       THIS       .D NOW^%DTC       .S FMDATE=%       .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")       .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")       .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN       TO SECOND       S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES       S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES       S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1)       S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1)       S:'\$D(COUNT) COUNT=0       K CHRJARR       S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL       I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45       .D CLM       .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS       I STYPE="A" D       .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR)       .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT       END EDIT FAIL)       .I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET       A2/20       .D CLM       .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS       .K CHRJARR S CHRJARR(0)=0           </pre>	

Routine Name	CHMXWB21
<pre> .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)=""  .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D ..I \$P(\$G(^CHMPAY(CHMPAY,0)), "^",2)=4 S CHRJARR(1,1)="F1*65*" ..I \$P(\$G(^CHMPAY(HACCLM,0)), "^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)), "^",4)&gt;CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^", 16) ..I \$P(\$G(^CHMPAY(CHMPAY,0)), "^",10)&gt;PAYDATE S PAYDATE=\$P(^0,"^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D .S CHMPAY=0,OK=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK ..I \$P(\$G(^CHMPAY(CHMPAY,0)), "^",2)=4 Q ..I \$P(\$G(^CHMPAY(CHMPAY,0)), "^",2)=0 Q ..S OK=0 ..Q Q OK  CLM ; N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACCT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" </pre>	

Routine Name	CHMXWB21
<pre> S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),",^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q </pre>	

### 3.2.1.9 Claim Adjudication/Payment Date for Denied Split Claim for Primary Claims

#### Functional Requirement: 2.8.1.9 Claim Adjudication/Payment Date for Denied Split Submission – Primary Claims

The CP&E system shall use the most recent Adjudication/Payment Date in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system and payment is denied for all claims.

Routine Name	CHMXWB21	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.1.9	
Related Options		
Related Routines	Routines “Called By”	Routines “Called”
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		

Routine Name	CHMXWB21
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)=" " S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI=" " D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)=" " .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY=" " D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,'\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1),"^",4),CHKEFT=\$P(^1),"^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0),"^",10) .I CHKDATE=" " N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE=" " N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q </pre>	



Routine Name	CHMXWB21
<pre> <b>CLM</b> ;       N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID       N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, <b>SUBNAME</b>       N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN <b>AMEX,PATDOB</b>       N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P <b>CIDNUM</b>       N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT       N PDI,CHMPAYI,REC,LN,STR       ; DO ACK AND PENDING       S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING"       S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4)       S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0))       D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES       S REC="" ; BUILD THE RECORD       F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D       .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR)       .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR)       W REC,! S REC="",COUNT=COUNT+1       Q </pre>	
<pre> GETCLMVAL       N CHCLB0,CHCLB1,CHMPAY1,CHCLE60       N TMPDTE S TMPDTE=0       S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))       S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))       S CHCLE64=\$G(^CHMXCLE(CHCLEI,64))       S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))       S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID       S BPPN=""       ; 6.BILLING PROVIDER PAYER NUMBER       S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID       S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME       S BPFNAME="" ;       9.BILLING PROVIDER FIRST NAME       S BPMNAME=""       ; 10.BILLING PROVIDER MIDDLE NAME       S BPNAMEX="" ;       11.BILLING PROVIDER NAME SUFFIX       S SPFTID=""       ; 12.RENDERING PHYSICIAN FEDERAL TAX ID       S SPPN=""       ; 13.RENDERING PHYSICIAN PAYER NUMBER       S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID       S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME       S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME       S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME       S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX       S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER       S EMPNAME="" ; 20.EMPLOYER NAME       S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))       S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID       S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME       S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME </pre>	

Routine Name	CHMXWB21
	<p>S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME  S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX  S PATID="" ; 26.PATIENT ID  S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME  S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME  S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME  S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX  S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH  S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER  S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN  S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER  S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)&gt;0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1))  ; 34.CHARGE AMT  <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b>  <b>AMOUNT</b>  <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b>  <b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b>  <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b>  S BTYPE=\$S(RUNTYPE="F":",",1:"") ; 39.BILL TYPE  S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; 40.PAYER CLAIM ID NUMBER (PDI)  S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) ; PDI NOT POPULATED,  DERIVED CLAIM CTL NUM  S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),"^",2) ; 41.PATIENT ACCOUNT NUMBER  S PRSCNUM=""  ; 42.PHARMACY PRESCRIPTION NUMBER  S VOUCHID="" ;  43.VOUCHER IDENTIFIER  S LOCSYSID=""  ; 44.APP/LOCATION SYSTEM ID  S GRPNUM=""  ; 45.GROUP NUMBER  S CLMTYP=\$GETCLMTYP ; GET  I/P/D CLAIM TYPE  S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47.  PROFESSIONAL SERVICE START DATE  S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1  INSTITUTIONAL START/END  S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2  DENTAL START/END  S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2)  Q</p>

### 3.2.1.10 Claim Adjudication/Payment Date for Paid Split Claim for Primary Claims

#### Functional Requirement: 2.8.1.10 Claim Adjudication/Payment Date for Paid Split Submission – Primary Claims

The CP&E system shall use the Adjudication/Payment Date from the most recent paid claim in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system.

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.1.10	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)	<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)=" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT </pre>	

Routine Name	CHMXWB21
<b>END EDIT FAIL)</b> .I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET <b>A2/20</b> .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)>0 LINE ; NO LINE <b>REJECTS RECORDED = NO LINE REPORTS</b> I STYPE="F" D . S (CHKDATE,CHKEFT,PAYDATE)="" . S CHRJARR(0)=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)>CHKDATE S CHKDATE=\$P(^("1","^",4),CHKEFT=\$P(^("1","^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)>PAYDATE S PAYDATE=\$P(^("0","^",10) . I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) . I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) . D CLM . D BLDSTC("CLM") . D LINE Q  <b>CLM ;</b> N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, <b>SUBLNAME</b> N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN <b>AMEX,PATDOB</b> N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P <b>CIDNUM</b> N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q  <b>GETCLMVAL</b> N CHCLB0,CHCLB1,CHMPAY1,CHCLE60 N TMPDTE S TMPDTE=0	

Routine Name	CHMXWB21
	S CHCLB0=\$G(^CHMXCLB(CHCLBI,0)) S CHCLB1=\$G(^CHMXCLB(CHCLBI,1)) S CHCLE64=\$G(^CHMXCLE(CHCLEI,64)) <b>S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))</b> S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID S BPPN="" ; 6.BILLING PROVIDER PAYER NUMBER S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME S BPFNAME="" ; 9.BILLING PROVIDER FIRST NAME S BPMNAME="" ; 10.BILLING PROVIDER MIDDLE NAME S BPNAMEX="" ; 11.BILLING PROVIDER NAME SUFFIX S SPFTID="" ; 12.RENDERING PHYSICIAN FEDERAL TAX ID S SPPN="" ; 13.RENDERING PHYSICIAN PAYER NUMBER S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER S EMPNAME="" ; 20.EMPLOYER NAME S CHCLC0=\$G(^CHMXCLC(CHCLCI,0)) S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX S PATID="" ; 26.PATIENT ID S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)>0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1)) ; 34.CHARGE AMT <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b> <b>AMOUNT</b> <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b> <b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$S(RUNTYPE="F":",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACT=\$P(\$G(^CHMXCLE(CHCLEI,0)), "^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER

Routine Name	<b>CHMXWB21</b>
S VOUCHID="" 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$\$GETCLMTYP I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q	
; GET ; 46/47. ; 46.1/47.1 ; 46.2/47.2	

## 3.2.2 Final Claim Status – Secondary Claims – CHAMPVA

### 3.2.2.1 Include Updated Claim Status for Secondary Claims in the Final Claim Status File

#### Functional Requirement: 2.8.2.1 Update Claim Status – Secondary Claims

The CP&E system shall update an electronic secondary CHAMPVA claim status to one of the following final claim statuses when received from FMS:

- F1 Finalized/Payment – VA has paid the claim/line
- F2 Finalized/Denial – VA has denied the claim/line

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.1	
Related Options		
Related Routines	Routines “Called By”	Routines “Called”
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	

Routine Name	<b>CHMXWB21</b>
Input Attribute Name and Definition	Name: Definition:
Output Attribute Name and Definition	Name: Definition:
Current Logic	
Same as 3.2.1.1	
Modified Logic (Changes are in bold and highlighted)	
<pre> CHMXWB21 ;HAC/DLB;WEB 5010 277 UNSOLICITED STATUS/DRIVER PART 1;10/25/10 ;;1.0;CHAMPVA SYSTEM;;JULY 11,2011;Build 1 ;;V1.0; ;;CALLED BY ^CHMXG001 TO PERFORM THE "ACK" STATUS ;;CALLED BY TASKMAN TO PERFORM THE "PENDING" AND "FINAL" STATUS ;;CALLS ENTRY POINTS TO CHMXWB22,CHMXWBUT ;;HR-COB-Medicare-A/B-Begin-9372 (27-May-2010)  ** ** ** 10/18/2011 ADDED TSTBLDCLM() ENTRY POINT TO VALIDATE THE REJECT VALUES BEING ** REPORTED IN THE CSTAT FILE ** 10/20/2011 MODIFIED THE LINE() FUNCTION TO FIX INCORRECT REJECT STATUS ** REPORTING IN THE CSTAT FILE ** 10/21/2011 ADDED THE SIX DATE GATHERING FUNCTIONS TO SUPPORT THE RULE ** PROVIDED BY BUSINESS GROUP FOR I/P/D CLAIMS ** 10/21/2011 ADDED THE GETCLMTYP() FUNCTION TO DETERMINE THE I/P/D CLAIM TYPE ** FOR BOTH 4010 AND 5010 INCOMING CLAIMS ** 10/24/2011 CHANGED THE DATERANGE FUNCTION TO A EXISTING DATE FUNCTION ** IN THE UTILITY ROUTINE: CHMXWBUT.INT ** 10/25/2011 CHANGED THE RENDERING PHYSICIAN NPI RETRIEVAL TO GET THE DATA ** FROM THE CHMXCLE(I,64) TIN DATA FIELD. ** 10/25/2011 ADDED A TEST FOR THE "U" VALUE RECEIVED FROM EMDEON TO RETURN "" ** IF NOT A "M" OR "F" (ONLY ACCEPTED VALUES) ** 10/26/2011 REMOVED ERROR MESSAGE OUTPUT FOR ERROR TRAP HANDLER ** ** ** 10/26/2011 CHANGED THE INSTITUTIONAL AND DENTAL START/END DATES FOR CLAIM ** LEVEL REPORTING TO GET START FROM ^CHMXCLE(I,1),FIELD 1 AND END ** TO ^CHMXCLE(I,1) FIELD 2 ** 11/4/2011 CHANGED THE DATE CHECK ROUTINE GETCLMI() TO RETURN THE CORRECT ** DATE FOR THE DESIRED FLAG ** ** ;1.0 IS THE INITIAL VERSION (HIPAA Ready LLC) ;2.0 UPDATES TO THE STATUS REPORT PER HIPAA II SPECIFICATIONS. ; *** Filenames should not be hard coded in files. See ^CHMX277 TRACKING ; GLOBAL FILE for the destination directory/filenames for the 5010 status ; files. ; Q  ***** ; PRODUCTION ENTRY POINTS TO PROCESS THE CURRENT STATUS. ; *****  EPACK(CHMXI) ; SET UP THE FILE/BATCH INDEX (PROVIDED BY CHMXG001) ; CHMXI I INDEX INTO ^CHMXCL() N COUNT,DIRFILE,GRPCNT,FILECNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA </pre>	

Routine Name	CHMXWB21
<p>TE,CHTPI,CHTPARR,CNT</p> <p>N MAXCNT,EICNT,CHA,CHMEDCOB,TPINT</p> <p>S FILECNT=0,RUNTYPE="A",GRPCNT=1,EICNT=0</p> <p>S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER</p> <p>D INIT</p> <p>S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA=""</p> <p>S CHTPI=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:CHTPI="" ; GET CLAIM T/P FOR FILE</p> <p>BEING CHECKED</p> <p>S CHTPI=\$O(^CHMXTP("C",CHTPI,"")) Q:CHTPI="" D ; GET "LOCAL" TP ID</p> <p>S IDX=\$O(^CHMX277("B",CHTPI,0))</p> <p>S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB"</p> <p>EMDEON PARTNER</p> <p>I CHMEDCOB=0 S TPINT=\$P(^CHMX277(IDX,0),"^")</p> <p>I TPINT="" S CHMEDCOB=+\$P(\$G(^CHMXTP(TPINT,0)),"^",11)</p> <p>D GETCHCLEI(CHMXI,CHTPI) ; RETRIEVE ^CHMXCLE INDEX</p> <p>Q:ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE</p> <p>D CREATEFILE</p> <p>D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION</p> <p>Q</p> <p>EPPENDTST</p> <p>N TEST S TEST=1</p> <p>D EPPEND</p> <p>Q</p> <p>EPPEND</p> <p>N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA</p> <p>TE,CHTPI,CHTPARR</p> <p>N CHNODEDEF,BREF,CHIIKVAL,DRDATA,TRKDATA,MAXCNT,CHMEDCOB,IDX</p> <p>; FILEMAN UPDATE VARIABLES</p> <p>N CHTO,CHFROM,TRKI,EICNT</p> <p>S RUNTYPE="P",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE</p> <p>D INIT</p> <p>S CHTPI="A"</p> <p>F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM</p> <p>"B" XREF</p> <p>.S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER</p> <p>.S IDX=\$O(^CHMX277("B",CHTPI,0))</p> <p>.S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB"</p> <p>EMDEON PARTNER</p> <p>.K CHTPARR</p> <p>.S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY</p> <p>.S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX</p> <p>.Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0)</p> <p>.S CHTPARR(CHTPI)="" ; SET CHTPARR UP FOR</p> <p>GETPENDEI</p> <p>.D GETTOFROM(CHTPI) Q:('CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL</p> <p>.D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES</p> <p>.Q:ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE</p> <p>.D CREATEFILE ; CREATE THE</p> <p>PENDING STATUS FILE</p> <p>.D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION</p> <p>Q</p> <p><b>EPFINAL ; entry point to start Final Status Creation file</b></p>	



Routine Name	CHMXWB21
<pre> N RUNTYPE,GRPCNT,EICNT,GROUPID,CHTPI,CHTPARR,CHFROM,CHTO,FILECNT N TRKI N COUNT,DIRFILE,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,FMDATE N CHNODEDEF,BREF,CHIJKVAL,DRDATA,TRKDATA,MAXCNT ; FILEMAN UPDATE VARIABLES S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .K CHTPARR S (CHFROM,CHTO,FILECNT,CHMEDCOB)=0 ; KILL TRADING PARTNER ARRAY; ASSUME NOT A "COB" TRADING PARTNER .S TRKI=\$O(^CHMX277("B",CHTPI,0)) ; GET TRACKING GLOBAL INDEX .I \$P(\$G(^CHMX277(TRKI,0)),"^",3)=0 Q .I \$P(^CHMX277(TRKI,0),"^",4)["COB" S CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .S CHTPARR(CHTPI)=" " .D GETTOFROM(CHTPI) Q:(^CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETFINALEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPREPTST N TEST S TEST=1 D EPREPLACEMENT Q  EPREPLACEMENT N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT,CHMEDCOB,IDX,CHTPX S RUNTYPE="H",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="REPLACEMENT" S TP=\$\$LISTTP^CHMXWB22 ; SELECT TRADING PARTNER S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER S IDX=\$O(^CHMX277("B",TP,0)) S:\$P(^CHMX277(IDX,0),"^",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHXI="" .F S CHXI=\$O(^CHMXCL("B",IDATE,CHXI)) Q:CHXI="" D ..S CHMXI=CHXI ..I \$D(TEST) W !,"CHECKING:",\$\$DTCVRT(\$E(IDATE,1,7)) ; SHOW FILE DATES BEING PROCESSED </pre>	

Routine Name	CHMXWB21
<pre> ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S CHTPX=\$P(\$G(^CHMXCLA(CHA,1)),",",1) Q:CHTPX="" ; GET CLAIM T/P FOR FILE BEING CHECKED ..S CHTPX=\$O(^CHMXTP("C",CHTPX,"")) ; CHECK SELECTED VS CLAIM TP ID ..I TP=CHTPX D ; FILE MATCHES THE SELECTED TP ..S CHTPI=TP ; CHTPI IS USED DOWNSTREAM ...I \$D(TEST) W " PROCESSING:",\$\$DTCVRT(\$E(IDATE,1,7)) ...I \$D(TEST) W !," EPREPLACEMENT: CHMXI= ",CHMXI," CHA= ",CHA," CHTPI= ",CHTPI," TP= ",TP ..S CHTPARR(TP)=" D GETCHCLEI(CHMXI,CHTPI) ; IF TP VALID, PROCESS FILE I \$D(TEST) W !,"REPLACE: \$J= ",\$J," REC COUNT= ",\$G(^ZSC(\$J,0))," CHTPI= ",CHTPI," TP= ",TP Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  DTCVRT(DATE) ; CONVERT DATE TO EXTERNAL FORMAT N EXTDATE S EXTDATE=\$E(EXTDATE,4,5)_"-"_ \$E(EXTDATE,6,7)_"-"_(\$E(EXTDATE,1,3)+1700) Q EXTDATE  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPOVERTST N TEST S TEST=1 D EPOVERLAY Q  EPOVERLAY N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT S RUNTYPE="O",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="OVERLAY" S TP=\$\$LISTTP^CHMXWB22 D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHMXI="" .F S CHMXI=\$O(^CHMXCL("B",IDATE,CHMXI)) Q:CHMXI="" D ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S TPTIN=\$P(\$G(^CHMXCLA(CHA,1)),",",1) Q:TPTIN="" ..I TP=\$O(^CHMXTP("C",TPTIN,"")) S CHTPARR(TP)=" D GETCHCLEI(CHMXI,TP) Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  ..... ; INIT PERFORMS LOCAL VARIABLE INITIALIZATION REQUIREMENTS FOR THE STATUS ; </pre>	

Routine Name	CHMXWB21
	<pre> ; PROCESSES. ; ..... INIT ; K ^ZSC(\$J) S ^ZSC(\$J,0)=0 K TMP S GROUPID="",CHTPI=0 I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S ATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND Q  ..... ; EPSTAT IS THE ENTRY POINT FOR THE ON-DEMAND REPORT FOR STATISTICS ; .....  EPSTAT D ONDEMAND^CHMXWB22 Q  EPREGEN D REGEN^CHMXWB22 Q  ..... ; FILE CREATION FUNCTIONS ; .....  CREATEFILE ; S MAXCNT=500000 ; HAPE POR 11/21/13 force max file size under 150 meg I RUNTYPE="F" S MAXCNT=175000 D FILELOOP Q  FILELOOP ; BUILD CLAIMS FROM ^ZSC(\$J) ARRAY (^CHMXCLE(I) VALUES) N CHCLEI,PDI D GETGRPCNT ; GET ESTIMATED GROUP FILE COUNT Q:GRPCNT=0 D GETFILE U DIRFILE ; CREATE ORIGINAL FILE D BLDHDR ; BLDHDR SETS COUNT=2 (SEE EMDEON SPEC) S CHCLEI=0 F S CHCLEI=\$O(^ZSC(\$J,CHTPI,CHCLEI)) Q:CHCLEI="" D .S EICNT=EICNT+1 .S STVAL=\$G(^ZSC(\$J,CHTPI,CHCLEI)) .D BLDCLM(CHCLEI,STVAL) ; GENERATE CLAIM/STC,DTL/STC RECORDS .D CHKSEQ ; CHECK FOR 500MB LIMIT, CREATE NEW FILE D BLDTRL ; BUILD THE TRAILER RECORD D CLOSEFILE ; CLOSE CURRENT FILE </pre>

Routine Name	CHMXWB21
<p><b>Q</b></p> <p>GETGRPCNT ; ESTIMATE THE TOTAL NUMBER OF FILES REQUIRED  N RECCNT,X  S RECCNT=0  S X=""  F S X=\$O(^ZSC(\$J,CHTPI,X)) Q:X="" S RECCNT=RECCNT+1  S GRPCNT=RECCNT\MAXCNT ; INTEGER DIVISION OF RECORD COUNT  S:RECCNT#MAXCNT&gt;0 GRPCNT=GRPCNT+1  ;W !,"GRPCNT= ",GRPCNT  Q</p> <p>CHKSEQ ; HANDLE THE FILE SEQUENCE NUMBERING REQUIREMENTS  I EICNT&gt;MAXCNT D  .D BLDTRL ; OUTPUT TRAILER RECORD  .D CLOSEFILE  .D GETFILE U DIRFILE  .D BLDHDR ; BLDHDR RESETS "COUNT" VARIABLE  Q</p> <p>FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN  Q:"APF"[RUNTYPE ; A=ACK, P=PENDING, F=FINAL  N CHNODEDEF,BREF,CHJKVAL,DRDATA,TRKDATA  S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_COUNT_"^.05="_CHMXI  T_"^.05="_CHMXI  S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_CHTO_"^.05="_CHFROM_"^.06="_COUNT  _"^.05="_CHFROM_"^.06="_COUNT  S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30,"TRKDATA=".02=1^.03="_DIRFILE_"^.04="_CHTO_"^.05="_CHFROM_"^.06="_COUNT  _"^.05="_CHFROM_"^.06="_COUNT  S BREF=FMDATE  S CHJKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE  S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT  D ADD^CHHRLIBFM(CHNODEDEF,CHJKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL  <b>I RUNTYPE="F" N X,X1 F S X=\$O(^ZSC1(\$J,X)) Q:X="" F S X1=\$O(^ZSC1(\$J,X,X1)) Q:X1="" I \$D(^CHMPAY("E",X,X1)) S ^CHMPAY("E",X,X1)=1</b>  Q</p> <p>.....  ;TEST ENTRY POINT TO TEST THE BLDCLM FUNCTION FROM A PDI VALUE  ; THERE ARE MULTIPLE CROSS REFERENCES (SOME DOCUMENTED, SOME UNDOCUMENTED  ; THAT ARE UTILIZED IN THE CLAIM PROCESSING SYSTEM. THE CVTPDI AND TSTBLDCLM  ; FUNCTIONS UTILIZE THE MAJORITY OF THESE CROSS REFERENCES.  .....</p> <p>CVTPDI(CHREFN)  N CHPCN,CHMXCLI,CHMXID,CHCLFI,SDATE,EDATE,CHCLFI,CHCLEI,IDX  I '\$D(^CHMXCLE("PDI",CHREFN)) W !,"NO ^CHMXCLE("PDI",CHREFN,") XREF" Q  S CHPCN=""  S1 S CHPCN=\$O(^CHMXCLE("PDI",CHREFN,CHPCN)) I CHPCN="" Q ;VALID PDI,NO DATA  S CHMXCLI=0  S2 S CHMXCLI=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI)) I 'CHMXCLI Q ;VALID PDI,NO DATA</p>	

Routine Name	CHMXWB21
S3	<pre> S CHMXID="" S CHMXID=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI,CHMXID)) I CHMXID="" Q S CHCLEI=\$P(CHMXID,"*",4) W !,"CLM SDATE [^CHMXCLE(I,1),1]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) W !,"CLM EDATE [^CHMXCLE(I,1),2]=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) S CHCLFI=0 W !,"LINE ITEM DATE VALUES" F IDX=1:1 S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI D ; CHECK ALL LINE ITEM DATES .W !,IDX," ", " L/I SDATE [^CHMXCLF(I,1),11]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) .W !," L/I END [^CHMXCLF(I,1),12]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) .W !," THERAPY SDATE [^CHMXCLF(I,2.5),1]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",1) .W !," LAST CERT DATE [^CHMXCLF(I,2.5),2]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",2) Q  TSTBLDCLM(CHPDI) N CHPCN,CHCLI,CHCLAI,CHCLEI,STVAL,COUNT,PDI,BUFI,RUNTYPE,CLMTYPE,TEST S (STVAL,CHPCN)="", (COUNT,CHCLEI,CHCLAI,PDI)=0,TEST=1 S CHPCN=0,CHPCN=\$O(^CHMXCLE("PDI",CHPDI,CHPCN)) I CHPCN="" W !,"NO CHPCN VALUE" Q E W !,"PATIENT CONTROL # [^CHMXCLE("PDI",CHPDI,PCN)]= ",CHPCN ; PATIENT CONTROL NUMBER S CHCLI=0,CHCLI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI)) I CHCLI="" W !,"NO CHCLI VALUE" Q S STVAL=0,STVAL=\$O(^CHMXCLE("A",CHCLI,STVAL)) W !,"XREF CLAIM STATUS= ",STVAL S BUFI="",BUFI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI)) I BUFI="" W !,"NO BUFFER INDEX VALUES" Q W !,"XREF [\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))] BUFFER INDICES= ",BUFI S CHCLEI=\$P(BUFI,"*",4) I CHCLEI="" W !,"NO CHCLEI VALUE RECOVERED" S RUNTYPE="H" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; RETRIEVE PDI S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) W !,"PDI VALUE FROM [^CHMXCLE(I,100),2 OR 4]= ",PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),^",1) W !,"X12 837 BUFFER [CHMXCL(I)= ",?30,CHCLI W !,"TRANSACTION [CHMXCLA(I)]= ",?30,CHCLAI W !,"PROVIDER BUFFER [CHMXCLB(I)]= ",?30,CHCLBI W !,"PATIENT BUFFER [CHMXCLC(I)= ",?30,CHCLCI W !,"CLM BUFFER [CHMXCLE(I)= ",?30,CHCLEI W !,"L/I BUFFER [CHMXCLF(I)]= ",?30,\$O(^CHMXCLF("B",CHCLEI,0)) S CLMTYPE=\$GETCLMTYP W !,"VERSION-FUNCTIONAL TYPE [^CHMXCLA(I,0),13]= ", \$P(\$G(^CHMXCLA(CHCLAI,0)),^",13) W !,"CLAIM TYPE= ", \$S(CLMTYPE="I": "INSTITUTIONAL",CLMTYPE="P": "PROFESSIONAL",CLMTYPE="D": "DENTAL") ; I/P/D CLAIM W !,"STATUS (ACK/PENDING)=", \$S(\$D(^CHMPAY("C",PDI))=0: "ACK",1: "PENDING") D CVTPDI(CHPDI) W ! D BLDCLM(CHCLEI,"") Q </pre>

Routine Name	CHMXWB21
<pre> ..... ; BLDCLM: ENTRY POINT TO BUILD THE CSTAT RECORDS ; THIS FUNCTION PROVIDES AN EXTERNAL ENTRY POINT TO GENERATE THE ; EMDEON CLAIM, CLAIM STATUS, DETAIL, AND DETAIL STATUS RECORDS ..... </pre>	
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1);TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1);GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0,"^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) </pre>	

Routine Name	CHMXWB21
	<pre> . D CLM . D BLDSTC("CLM") . D LINE Q  GETSTYPER(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D . S CHMPAY=0,OK=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)),^",2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)),^",2)=0 Q .. S OK=0 .. Q Q OK  ..... ; FUNCTIONS TO GATHER INFORMATION REGARDING REJECT REASONS ; .....  BLDACK ; POPULATE CHRJARR REJECT REASON/REJECT CODE VALUES K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY S REJCODES="" D GTRXRJRSN^CHMXWBUT(CHCLAI) ; TRANSACTION "101" NODE REJECT REASONS D GPRORJRSN^CHMXWBUT(CHCLBI) ; PROVIDER "101" NODE REJECT REASONS D GPATRJRSN^CHMXWBUT(CHCLCI) ; PATIENT "101" NODE REJECT REASONS D GCLMRJRSN^CHMXWBUT(CHCLEI) ; CLAIM "101" NODE REJECT REASONS I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" ;I \$D(TEST) W !,"BLDACK: RJCNT=",CHRJARR(0)," RJARR(1,1)=",CHRJARR(1,1) Q  ..... ; RECORD BUILDING FUNCTIONS: \$TEXT FILES PROVIDE FIELD DEFINITIONS ; .....  BLDHDR ;BUILD HEADER RECORD N LN,REC,STR,LOADTYPE S (STR,LN,REC)="",COUNT=2 ; EMDEON SPEC: REC. COUNT STARTS @ 2 D GETHDRVAL ; SET UP HEADER VARIABLES F LN=1:1 S STR=\$T(EMDEONHDR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="" Q  CLM ; </pre>

Routine Name	CHMXWB21
<pre> N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDY,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q  LINE ; BUILD DTL RECORDS FOR LINE ITEMS N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D .D GDTLVAR .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),^") .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1" Q ... S CHRJARR(1,1)="F0*1" ... Q .. Q .D BLDSTC("DTL") Q  BLDSTC(TYPE) ; WRITE OUT THE STC RECORDS FROM CHRJARR REJECT ARRAY ; TYPE CLAIM OR DETAIL LEVEL EMDEON STATUS VALUES &amp; LINE ITEM CONTROL NUMBER N PCINUM,LICNUM,DATERR,ESCODE N IDX,JDX,REJCODES,ECNT N LN,STR,REC S REJCODES=0 </pre>	



Routine Name	CHMXWB21
<pre> S ECNT=CHRJARR(0) ; FRONT END EDIT REJECT COUNT F IDX=1:1 Q:(IDX&gt;ECNT) D          ; PROVIDE STC RECORD FOR EACH ERROR ..F JDX=1:1 Q:\$G(CHRJARR(IDX,JDX))="" D  ; GET EACH REJECT VALUE STORED ..S (RJCODE,RJSTATUS,ENTITY)="" ..D GETSTCVAL(TYPE) ; EMDEON STATUS RECORD VALUES FOR CLM OR DTL ..S REJCODES=\$G(CHRJARR(IDX,JDX)) ..;W:\$D(TEST) !,"BLDSTC: REJCODE STR=",REJCODES,! ; DEBUG: ONLY OUTPUTS IF "TEST" IS DEFINED ..S RJCODE=\$P(REJCODES,"*",1),RJSTATUS=\$P(REJCODES,"*",2),ENTITY=\$P(REJCODES, "*,3) ..S REC="" ..F LN=1:1 S STR=\$T(EMDEONSTC+LN) Q:STR["END OF RECORD" D          ; GENERATE THE STATUS (STC) RECORD ...I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ...E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 Q  BLDTRL                                     ; TRAILER RECORD N LN,REC,STR S COUNT=COUNT-2 ; ADJUST COUNT FOR START OF 2 (SEE EMDEON SPEC) S (STR,LN,REC)="" F LN=1:1 S STR=\$T(EMDEONTRLR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ; 10/20/2010 ADDED "ELSE" W REC,! S REC="" Q  ..... ; GENERIC FUNCTIONS SUPPORTING THE RECORD BUILDING ; .....  GETFILE() N DIRPATH S EICNT=0,FILECNT=FILECNT+1 S DIRPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP D NOW^%DTC S FMDATE=% S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT" E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT" ; CHMEDCOB SET IN CHMXWT1 E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME I RUNTYPE="H" S DIRFILE=DIRFILE_"_H" I RUNTYPE="O" S DIRFILE=DIRFILE_"_O" S FLAG=\$\$OFILE^CHMXWBUT(DIRFILE,"NWS") I 'FLAG D VMSERR Q "" S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q </pre>	

Routine Name	CHMXWB21
<pre> TST     N FMDATE,TIME     S FMDATE=3110901.1234     W !,FMDATE     S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")     W !,FMDATE     S TIME=\$E(FMDATE,9,14)     W !,TIME     S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")     W !,TIME     Q  CLOSEFILE     C DIRFILE     Q  GETCHCLEI(CHCLI,CHTPI) ; BUILD ARRAY OF CLAIM INDICES (CHMXCLE(I) VALUES)     ; CHCLI I INDEX FOR ^CHMXCL() FILE (CLAIM FILE RECEIVED)     N STAT,CHCLAI,PDI,CHIDX,CHCLEI,RECCNT,CTPI S CTPI=0     S STAT="",STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) I \$D(TEST) W:STAT="" !,"NULL STAT" Q:STAT="" ; RETRIEVE STATUS VALUE     S CHCLAI="",CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) I \$D(TEST) W:CHCLAI="" !,"NULL CHCLAI" Q:CHCLAI="" ; RETRIEVE THE ^CHMXCLA(I) VALUE     I \$D(TEST) W !," GETCHCLEI: READING:",CHCLI," STATUS= ",STAT," ^CHMXCLA(I)= ",CHCLAI," TP= ",CTPI     S STAT=""     F S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) Q:STAT="" D     .S CHCLAI=""     .F S CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) Q:CHCLAI="" D     ..S PDI=0     ..F S PDI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI)) Q:PDI D     ...S CHIDX=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,0))     ...Q:CHIDX="" ; PDI RETRIEVES BUFFER IDXs     ...S CHCLEI=\$P(CHIDX,"*",3)     ...S ^ZSC(\$J,CHTPI,CHCLEI)=STAT,RECCNT=\$I(^ZSC(\$J,0))     Q  GETTP(CHA)     N TPID,ID     S ID=""     S TPID=\$P(\$G(^CHMXCLA(CHA,1)), "^",1)     S:TPID]"" ID=\$O(^CHMXTP("C",TPID,""))     Q IDER  GETPENDEI(CHFROM,CHTO) ; GET CHCLEI FROM PENDING ^CHMPAY(I) FROM/TO VALUES     K ^ZSC(\$J)     N IDX,CHJK,PDI,CHCLEI,CHCLAI,CTPI,RECCNT S RECCNT=0     F IDX=CHFROM:1 Q:IDX&gt;CHTO D     .S CHJK=\$O(^CHMPAY(IDX,"PDI",0)) Q:CHJK=""     .S PDI=\$P(^CHMPAY(IDX,"PDI",CHJK,0),"^",1) Q:PDI=""     .S CHCLEI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",6) Q:CHCLEI=""     .S CHCLAI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",3) Q:CHCLAI=""     .S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM </pre>	

Routine Name	CHMXWB21
<pre> .I CHCLEI]" I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1 S ^ZSC(\$J,0)=RECCNT Q  <b>GETFINALEI(CHFROM,CHTO) ; GET CHCLEI FOR FINAL CLAIM STATUS</b>   K ^ZSC(\$J),^ZSC1(\$J)   N CHCLEI,CHCLAI,CTPI,RECCNT,CLC,CLB,CHPIDX,CHPDATE,CHPPDI   S RECCNT=0   S %H=\$H-180 D YMD^%DTC S CHPDATE=X   F S CHPDATE=\$O(^CHMPAY("E",CHPDATE)) <b>Q:CHPDATE="" S CHPIDX="" F S CHPIDX=\$O(^CHMPAY("E",CHPDATE,CHPIDX))</b> <b>Q:CHPIDX="" I \$G(^CHPIDX))'=1 D</b>   . S CHPPDI=\$O(^CHMPAY(CHPIDX,"PDI",0)) Q:CHPPDI=""   . I CHPPDI=\$P(\$G(^CHMPAY(CHPIDX,"PDI",PDI,0)),")^")   . I CHPPDI="" Q   . S CHCLEI=\$O(^CHMXCLE("D",CHPPDI,"")) I CHCLEI="" Q   . I \$\$FINAL(CHCLEI) D   .. S CLC=\$P(^CHMXCLE(CLE,0),")^")   .. S CLB=\$P(^CHMXCLC(CLC,0),")^")   .. S CHCLAI=\$P(^CHMXCLB(CLB,0),")^")   .. S CTPI=\$\$GETTP(CHCLAI) ; <b>TRADING PARTNER FOR CURRENT CLAIM</b>   .. I \$D(CHTPARR(CTPI)) <b>S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1,^ZSC1(\$J,CHPDATE,CHPIDX)=""</b> <b>S ^ZSC(\$J,0)=RECCNT</b> <b>Q</b>  ..... ; GETCTPI RETURN THE NEXT TRADING PARTNER INDEX BASED ON THE LAST INDEX ; ; USED. THIS FUNCTRION IS CALLED BY THE "PENDING STATUS" PROCESS. ; ; FUNCTION USES LOCAL VARUIABLE "CHTPI" AS THE BASIS FOR RETURNING ALL ; ; TRADING PARTNER INDICES. IF TRADING PARTNER IS INACTIVE, FUNCTION MOVES ; ; TO THE NEXT TRACKING GLOBAL TRADING PARTNER. ; .....  GETCTPI   N TRKI,CHTPI   S CHTPI=""   F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF   .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX   .I (\$P(\$G(^CHMX277(TRKI,0)),")^",3)=1) S CHTPARR(CHTPI)="   Q  GETCLMI(,DATE,FLAG) ; MATCH A DATE WITH THE ^CHMXCL(I) ENTRY ; DATE DATE FOR THE ^CHMXCL() SEARCH ; FLAG "&gt;","&lt;","=" OR "&gt;" DATE SEARCH POSSIBILITIES N CHCNT,LSTI,EXIT,WDATE,RTN S RTN=0,EXIT=0 S LSTI=(\$P(^CHMXCL(0),")^",3))+1 F CHCNT=1:1 Q:(LSTI="")!(EXIT) S LSTI=\$O(^CHMXCL(LSTI),-1) D .S WDATE=\$P(\$P(\$G(^CHMXCL(LSTI,0)),")^",1),") ; FILE OPENED DATE .I ((FLAG="&lt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &gt; SPECIFIED DATE .E I ((FLAG="=")&amp;(WDATE=DATE)) S RTN=LSTI,EXIT=1 ; DATE = OR &gt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; OR = SPECIFIED DATE Q RTN </pre>	

Routine Name	CHMXWB21
	<pre> ..... ; FILEMAN UPDATE "DR" STRING BUILD FUNCTION ; .....  SETDARR(STR)   N REG,DRDATA,TMP,IDX,DARR   S DARR="",DRDATA=""   F IDX=1:1 S TMP=\$P(STR,"^",IDX) Q:TMP="" D   .S REG=\$P(TMP,"=",1),VAL=\$P(TMP,"=",2)   .S DARR(REG)=VAL   S DRDATA=\$\$SETDR^CHHRLIBFM("DARR")   Q DRDATA  ..... ; RETRIEVE VALUES FROM THE TRACKING GLOBAL ; ; NEEDS: CHTPI: TRADING PARTNER INDEX TO USE ; ; RUNTYPE: STATUS TYPE BEING CREATED ; .....  GETTOFROM(CHTPI) ; RETRIEVE TO-FROM FILES FROM TRACKING GLOBAL   N JDX,NODE,TRKI   S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; CONVERT CHTPI FOR TRACKING GLOBAL ACCESS   S NODE=\$S(RUNTYPE="P":20,RUNTYPE="F":30) ; SELECT PENDING/FINAL STATUS NODE   S JDX="A",JDX=\$O(^CHMX277(TRKI,NODE,JDX),-1) ; LOOK AT LAST ENTRY FOR THE STATUS   ;W !,"GETTOFROM: CHTPI = ",CHTPI," NODE= ",NODE," JDX= ",JDX   Q:JDX=""   S CHFROM=(\$P(\$G(^CHMX277(TRKI,NODE,JDX,0)),"^",4)) ; NEW "FROM"= OLD "TO"   S:CHFROM'=0 CHFROM=CHFROM+1,CHTO=\$P(\$G(^CHMPAY(0)),"^",3) ; SET START/ END INDEX   ;W !,"GETTOFROM: FROM= ",CHFROM," TO= ",CHTO   Q  ..... ; SUPPORT FUNCTIONS FOR THE FILE GENERATION FUNCTIONS DEFINED ABOVE. ; .....  DATERANGE(FROM,TO) ;Prompt for a FROM and TO date range.   N MAXTO   D NOW^%DTC S MAXTO=\$E(%,1,7)   S (FROM,TO)="", (DONE,POP)=0   F Q:(DONE)!(POP) D   .S FROM=\$\$GETDATE^CHMXWBUT("ENTER START DATE: ") S:FROM=-1 POP=1 Q:POP   .I ((FROM&lt;1)!(FROM&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S TO=\$\$GETDATE^CHMXWBUT("ENTER END DATE: ") S:TO=-1 POP=1 Q:POP   .I ((TO&lt;FROM)!(TO&gt;MAXTO)) W " [Invalid date. Try again]" Q   .S DONE=1   Q POP  GETSTVAL(CHCLEI) ; REVERSE POLISH METHOD FOR RETRIEVING CLAIM STATUS   N CHCLCI,CHCLBI,CHCLAI,CHCLI,TARGEI,STAT,PDI   S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES </pre>

Routine Name	CHMXWB21
<pre> S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S PDI=\$P(^CHMXCLE(CHCLEI,100),"^",2) S:PDI="" PDI=\$P(^CHMXCLE(CHCLEI,100),"^",4) I PDI="" Q 6 ;Assume 6 that it was rejected if no PDI found...data integrity issue S STAT=0,TARGEI=0 F Q:TARGEI=CHCLEI D .S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) .S CHIDXS=0,CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,"")) .S TARGEI=\$P(CHIDXS,"*",3) Q STAT  ..... ; EMAIL STATUS TO INTERESTED PARTIES ; .....  AUTOMM(TYPE) D NOW^%DTC S FMDATE=\$E(%,1,7) S CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL_ (Total Records = "_CHBRCCNT_") S CHNB=CHNB+1,ZML(CHNB)="" I TYPE="EFILE" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: ***ERROR*** CREATING 277U STAT file!" E I TYPE="OK" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: Successful CREATION 277U STAT file!" S CHNB=CHNB+1,ZML(CHNB)="EDI BATCH NUMBER ^CHMXCL("_CHMXI_",0)" S XMTEXT="ZML(",XMSUB="277U WebMD..Success.."_FMDATE S XMDUZ=.5 S XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="" S XMY("FRANK.PELUSO@VA.GOV")="" D ^XMD Q  ..... ; THE FOLLOWING ROUTINES ARE PRESERVED FOR HISTORICAL PURPOSES ; ; THERE ARE COPIES OF THESE FUNCTIONS IN CHMXWB07,CHMXWB11,CHMXWB21 ; .....  ; EXFTP S CHFTPMSG=0 ;H 1000 ;DEF4399 ajm removed hang, not needed XCOM S X=\$ZF(-1,"SUBMIT DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP.COM /NAME=WEB_WS_FTP_JOB/NOPRINTER/USER=DSMMANAG/PARAM="_CHFILE) ;//////// UNREMARK/REMARK OUT CODE FOR PRODUCTION ////////// ;H 1200 ;DEF4399 ajm removed hang, not needed K ^CHMZHOLD("RKN_W277") Q ;DEF4399 ajm addd - files are copied, no longer ftp'd ; FTPCK S RD="DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP_JOB.LOG" ;LIVE C RD K CHFTPTIM,CHFTPWT S ZE="",QFLAG=0,CHFTPWT=\$P(^CHMDIC(741002.17,1,2),"^",10) F CHFTPTIM=1:1:CHFTPWT H 60 O RD:"R" D Q:(QFLAG=1) ;RKN, 6-29-05, MOD READONLY .I \$T C RD Q </pre>	

Routine Name	CHMXWB21
<pre> .F U RD R RDLIN D Q:(QFLAG=1)!(\$ZE["ENDOFIL"]) ..I (RDLIN["226 Transfer complete") S CHFTPMSG=1,QFLAG=1 Q ..I (RDLIN["Charged CPU time:") S QFLAG=1 Q FTPERR C RD S CHNB=2 S ZML(CHNB)=" S ZML(CHNB)=CHVMSFL S CHNB=CHNB+1 S ZML(CHNB)=" S CHNB=CHNB+1 S XMTEXT="ZML(" K CHFTPMSG,RD,RDLIN ; ; ; ; CALCULATE A DATE 90 DAYS IN THE PAST ; ; ; DOLDFLS D NOW^%DTC S X1=X,X2=-90 D C^%DTC D YX^%DTC Q ;  ; ; Error Handler / personnel notification functions for the 277 status file ; ; generation process. ; ;  VMSERR C IO D ^%ZTER ; ERRMM ; ;S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)="",CHNB=CHNB+1 ;S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 ;S ZML(CHNB)=" 277U WebMD file!" ;S XMDUZ=.5,XMY("10722")="" ;RKN LIVE ;S XMTEXT="ZML(",XMSUB="277U WebMD..Error.."_FMDATE S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)= "",CHNB=CHNB+1 S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 S ZML(CHNB)=" CSTAT file!" ;S XMDUZ=.5,XMY("JAMES.NELSON5@VA.GOV")="" ;RKN LIVE ;S XMY("SREEDHAR.MADANAPALLE@MED.VA.GOV")="" S XMDUZ=.5,XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="",XMY("FRANK.PELUSO@VA.GOV")="" S XMTEXT="ZML(",XMSUB="277U..Error.."_DATESTAMP ; BAS -END MOD D ^XMD Q </pre>	

Routine Name	CHMXWB21
<pre> ..... ; \$TEXT VARIABLE SETUP ROUTINES ; ..... </pre>	
<pre> GETHDRVAL   S LOADTYPE=\$S(RUNTYPE="H":"F",1:"I")   S (PAYNAME,PAYPHONE)=" <b>I RUNTYPE="F" S PAYNAME="CHAMPVA CUSTOMER</b> <b>SERVICE",PAYPHONE="8007338387"</b>   Q </pre>	
<pre> GETCLMVAL   N CHCLB0,CHCLB1,CHMPAY1,CHCLE60   N TMPDTE S TMPDTE=0   S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))   S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))   S CHCLE64=\$G(^CHMXCLE(CHCLEI,64)) <b>S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))</b>   S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID   S BPPN=""   ; 6.BILLING PROVIDER PAYER NUMBER   S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID   S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME   S BPFNAME="" ;   9.BILLING PROVIDER FIRST NAME   S BPMNAME=""   ; 10.BILLING PROVIDER MIDDLE NAME   S BPNAMEX="" ;   11.BILLING PROVIDER NAME SUFFIX   S SPFTID=""   ; 12.RENDERING PHYSICIAN FEDERAL TAX ID   S SPPN=""   ; 13.RENDERING PHYSICIAN PAYER NUMBER   S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID   S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME   S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME   S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME   S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX   S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER   S EMPNAME="" ; 20.EMPLOYER NAME   S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))   S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID   S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME   S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME   S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME   S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX   S PATID="" ; 26.PATIENT ID   S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME   S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME   S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME   S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX   S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH   S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER   S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN   S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER </pre>	

Routine Name	CHMXWB21
S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)>0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1)) ; 34.CHARGE AMT <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b> <b>AMOUNT</b> <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b> <b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$S(RUNTYPE="F":",",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),"^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER S VOUCHID="" ; 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47. PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1 INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2 DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q GPSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S STDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11),TMPDTE=STDATE ; TMPDTE IS LAST START DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:CHFI D ; CHECK ALL LI "FROM" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11) .I TDATE<STDATE S STDATE=TDATE ; TRACK "OLDEST" START DATE .I TDATE>TMPDTE S TMPDTE=TDATE ; TRACK "MOST RECENT" START DATE S EDATE=\$\$GVCEND S:(EDATE="")!(EDATE<TMPDTE) EDATE=TMPDTE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES GISTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL CLAIM START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S CHFI=0 S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) ; END DATE S SVCDATES=STDATE_"*"_EDATE	



Routine Name	CHMXWB21
<p>Q SVCDATES</p> <p>GDSTARTEND() ; POPULATE EMDEON FIELD FOR DENTAL CLAIM START/END DATES  N STDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2) ; END DATE  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>GETSTCVAL(TYPE)  S PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 4.PAYER CLAIM ID NUMBER(PDI)  S:PCINUM="" PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED,  DERIVED CLAIM CTL NUM  S DATERR="" ; 10.EMDEON SPECIFIC DATA IN ERROR  S ESCODE="" ; 11.EMDEON SPECIFIC STATUS CODE  I RUNTYPE="F" D Q  .I \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICNUM=\$P(^",23)  .I \$G(LICNUM)= "" S LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2)  S:TYPE="CLM" LICNUM="" ; 5.CLM LVL LINE ITEM CONTROL NUMBER  S:TYPE="DTL" LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; SERVICE LINE NUMBER  Q</p> <p>.....  ; 9/7/2011: modified Procedure Modifier #1 extract to field 4 from field 3;  .....</p> <p>GDTLVAR\$ ; DTL RECORD DATA GATHERING  S LICN=""  S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER  S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED  CLAIM CTL NUM  S CHCLF1=\$G(^CHMXCLF(CHCLFI,1))  I RUNTYPE="F", \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICN=\$P(^",23)  I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER  S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID  S:SQID="" SQID="NU" ;  DEFAULT VALUE IF NOT PROVIDED  S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE  S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT  PROVIDED  S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1  S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2  S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3  S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4  S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT  S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL  LINE ITEM CHARGE AMOUNT  S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND  LINE ITEM PROV. PAYMENT AMOUNT  S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE  ITEM PROV. PAYMENT AMOUNT  S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE  S:RVNUCODE="9999" RVNUCODE="" ; DON'T  OUTPUT 9999 VALUE  S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF</p>	

Routine Name	CHMXWB21
<p>SERVICE)</p> <p>S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL</p> <p>QUANTITY(UNITS OF SERVICE)</p> <p>S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 16.EMDEON CLAIM NUMBER</p> <p>S CLMTYP=\$\$GETCLMTYP ; GET</p> <p>I/P/D CLAIM TYPE</p> <p>S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17.</p> <p>PROFESSIONAL SERVICE START/END DATE</p> <p>S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1</p> <p>INSTITUTIONAL SERVICE START/END DATES</p> <p>S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL</p> <p>SERVICE START/END DATES</p> <p>S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2)</p> <p>Q</p> <p>.....</p> <p>; LINE ITEM FUNCTIONS TO GATHER THE SPECIFIC DATA FIELDS ;</p> <p>.....</p> <p>.....</p> <p>; EMDEON REQUIRES THE EARLIEST START DATE (CLAIM OR LINE ITEM DATE ;</p> <p>; FOR THE SERVICE START DATE. THIS FUNCTION 1) GETS THE CLAIM START ;</p> <p>; DATE, THEN SEARCHES THROUGH LINE ITEMS FOR ANY EARLIER DATES. ;</p> <p>; THE EARLIEST DATE FOUND IS RECORDED AS THE START DATE. ;</p> <p>.....</p> <p>.....</p> <p>; DETERMINE THE CLAIM TYPE, I.E. PROFESSIONAL, INSTITUTIONAL, OR DENTAL</p> <p>; 4010/5010 VERSION IDENTIFIERS</p> <p>; 98 / 222 PROFESSIONAL CLAIMS</p> <p>; 96 / 223 INSTITUTIONAL CLAIMS</p> <p>; 97 / 224 DENTAL CLAIMS</p> <p>.....</p> <p>.....</p> <p>GETCLMTYP()</p> <p>N TYPE ; VERSION-FUNCTIONAL TYPE NUMBER FROM ^CHMXCLA()</p> <p>S TYPE=\$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13)</p> <p>Q:(TYPE[222]!(TYPE[98] "P"</p> <p>Q:(TYPE[223]!(TYPE[96] "I"</p> <p>Q:(TYPE[224]!(TYPE[97] "D"</p> <p>Q "U"</p> <p>.....</p> <p>; PROFESSIONAL START/END DATES</p> <p>.....</p> <p>GPDSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL START/END</p> <p>DATES</p> <p>N STDATE,CHFI,TDATE,EDATE,SVCDATES</p> <p>S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE</p> <p>S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE</p> <p>S:EDATE="" EDATE=STDATE</p> <p>S SVCDATES=STDATE_ "*" _EDATE</p> <p>Q SVCDATES</p>	

Routine Name	CHMXWB21
<p>GIDSTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL DETAIL START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11)  S:STDATE="" STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12)  S:EDATE="" EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11)  S:EDATE="" EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),^",2)  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>GDDSTARTEND() ; EMDEON DENTAL DETAIL START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) ; START DATE  S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) ; END DATE  S:EDATE="" EDATE=STDATE  S SVCDATES=STDATE_ "*" _EDATE  Q SVCDATES</p> <p>.....  ; EMDEON REQUIRES THE LAST END DATE (CLAIM OR LINE ITEM DATE ;  ; FOR THE SERVICE END DATE. THIS FUNCTION 1) GETS THE CLAIM START ;  ; DATE, THEN 2)SEARCHES THROUGH LINE ITEMS FOR ANY LATER DATES. ;  ; THE LATEST DATE FOUND IS RECORDED AS THE END DATE. ;  .....</p> <p>GSVCEND() ; POPULATE EMDEON REQUIRED FIELD FOR SERVICE END DATE  N ENDATE,CHFI,TDATE  S CHFI=0,CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI))  S ENDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12) ; ^CHMXCLF(I) END DATE  F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:CHFI D ; CHECK ALL LI "TO" DATES  .I TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12)  .I TDATE&gt;ENDATE S ENDATE=TDATE ; TRACK "YOUNGEST" END  DATE  Q ENDATE</p> <p>GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT  Q 0</p> <p>GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT  Q 0</p> <p>GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE)  Q 0</p> <p>.....  ; DOCUMENTREC (UTILITY THAT HELPS IN DOCUMENTING THE RECORD GENERATION  PROCESS) ;  ; CREATES A FILE THAT CONTAINS THE RECORD INFORMATION FOR THE 5010 ;  ; EMDEON STATUS RECORDS, INCLUDING THE HEADER, CLAIM, LINE ITEM, AND  TRAILER ;  ; RECORDS. THIS FUNCTION USES THE FIELD DESCRIPTORS TO DOCUMENT EACH  FIELD IN ;  ; THE RECORDS, I.E.: ;  ; 1) RECORD NAME ;  ; 2) STARTING LOCATION IN THE RECORD ;</p>	

Routine Name	CHMXWB21
<pre> ; 3) LENGTH (WIDTH) OF THE FIELD ; ; 4) JUSTIFICATION WITHIN THE FIELD ; ; 5) THE VALUE (HARD CODED FIELDS) OR THE CACHE FILELOCATION FROM WHICH THE ; ; VALUE IS RETRIEVED. ; ..... </pre>	
<p>DOCUMENTREC</p> <pre> N DOCFILE,TMPIO,LN,HTABS,FTABS,RTYPE,RNAME,CHTYPE,STR,COLNMS S RTYPE="EMDEONHDR^EMDEONCLM^EMDEONSTC^EMDEONTRLR^EMDEONLI" ; NAMES OF \$TEXT TABLES S COLNMS="FIELD NAME^USE^DESC^PAD^JUST^VALUE" ; COLUMN HEADER </pre>	
<p>NAMES</p> <pre> S HTABS="35^39^45^50^57" ; HEADER TAB STOPS FOR THE FIELD </pre>	
<p>DESCRIPTIONS</p> <pre> S FTABS="36^39^46^52^57" ; FIELD TAB STOPS FOR DESCRIPTIONS S DOCPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" S DOCFILE="DOC277_5010_"_DATESTAMP_".TXT" ; STATUS MAPPING </pre>	
<p>DOCUMENTATION</p> <pre> S DOCFILE=DOCPATH_DOCFILE ; OUTPUT THE \$TEXT </pre>	
<p>TO A PRINTABLE FILE</p> <pre> S FLAG=\$\$OFIL^CHMXWBUT(DOCFILE,"NWS") S TMPIO=\$IO U DOCFILE F CHTYPE=1:1 S RNAME=\$P(RTYPE,"^",CHTYPE) Q:RNAME="" D .W !!,?20,"HEALTH CARE CLEARING HOUSE """,RNAME,"" RECORD DEFINITIONS" ; TITLE .W !,\$P(COLNMS,"^",1),?P(HTABS,"^",1),\$P(COLNMS,"^",2),?P(HTABS,"^",2),\$P(COLNMS,"^",3),?P(HTABS,"^",3),?P(COLNMS,"^",4),?P(HTABS,"^",4),\$P(COLNMS,"^",5),?P(HTABS,"^",5),\$P(COLNMS,"^",6),?P(HTABS,"^",6),?P(COLNMS,"^",7) .F LN=1:1 S STR=\$T(@RNAME+LN) Q:STR["END OF RECORD" D ; READ \$TEXT </pre>	
<p>DESCRIPTOR</p> <pre> ..W !,\$P(STR,";",3),?P(FTABS,"^",1),\$P(STR,";",10),?P(FTABS,"^",2),\$P(STR,";",8),?P(FTABS,"^",3),?P(STR,";",7),?P(FTABS,"^",4),\$P(STR,";",6),?P(FTABS,"^",5),\$P(STR,";",4) </pre>	
<pre> U TMPIO D CLOSEFILE^CHMXWBUT(DOCFILE) ; CLOSE CURRENT FILE Q ..... ; EMDEONHDR: Common Header for EMDEON STATUS Files ; ; A single header is generated for each output file. ; ; DESC: "FIELD NAME";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA TYPE"; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR(record # and text description) ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT, C= CENTER ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH (ANY PRINTABLE CHARACTER) ; ; NOTE: PAD CHAR="" IF NO CHARACTER IS BETWEEN THE SEMICOLONS (I.E. ;;) ; ; NO PADDING WILL OCCUR IF THIS IS SET UP THIS WAY ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; ; ; FORMATDATA TREATS THE PAD CHAR (:) AS A NULL, SO NO PADDING OCCURS. ; ; THIS WILL ALLOW USE OF THE FORMATDATA FUNCTION WITHOUT MODIFICATION </pre>	

Routine Name	CHMXWB21
<p>BETWEEN ;</p> <p>    ; PADDED AND NON-PADDED FIELDS. ;</p> <p>.....</p> <p>    ; 8/4/11 DLB "6. CREATION TIME" FROM \$(DATESTAMP,9,12) TO \$(DATESTAMP,9,14) ;</p> <p>    ; 8/15/11 DLB "2. FILE GROUP ID" INSERTED THE DATESTAMP VALUE TO ENSURE</p> <p>UNIQUENESS ;</p> <p>    ; 9/7/2011 DLB 12. LOAD TYPE CHANGED TO PROVIDE "F" WHEN HISTORICAL FILE</p> <p>GENERATED ;</p> <p>.....</p> <p>EMDEONHDR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";DATA</p> <p>LENGTH/PATTERN";FIELD START LOCATION;FIELD USE</p> <p>    ;;1.RECORD ID;"HDR";3;L;;3AN;0;R;</p> <p>    ;;2.FILE GROUP ID;\$(GROUPID="'":GROUPID,1:DATESTAMP);20;L;;20AN;4;R;</p> <p>    ;;3.FILE GROUP SEQUENCE NUMBER;FILECNT;2;R;0;3N;24;R;</p> <p>    ;;4.FILE GROUP COUNT;GRPCNT;2;R;0;3N;26;R;</p> <p>    ;;5.CREATION DATE;\$(DATESTAMP,1,8);8;L;;8AN;28;R;</p> <p>    ;;6.CREATION TIME;\$(DATESTAMP,9,14);6;L;;6N;36;R;</p> <p>    ;;7.TRADING PARTNER ID;"VAFNH";10;L;;10AN;42;R;</p> <p>    ;;8.SUBMITTER NAME;"VA HEALTH ADMIN CENTER";30;L;;30AN;53;R;</p> <p>    ;;9.PAYER CONTACT NAME;PAYNAME;60;L;;60AN;83;O;</p> <p>    ;;10.PAYER SUPPORT TELEPHONE NUMBER;PAYPHONE;10;L;;10N;143;O;</p> <p>    ;;11.PAYER SUPPORT EMAIL ADDRESS;"";80;L;;80AN;153;O;</p> <p>    ;;12.LOAD TYPE;LOADTYPE;1;L;;1AN;233;R;</p> <p>    ;;13.PAYER UNIQUE FILE IDENTIFIER;DATESTAMP;20;L;;20AN;234;R;</p> <p>    ;;14.FILE TYPE;"CStat";5;L;;5AN;254;R;</p> <p>    ;;15.VERSION CODE;"03";2;L;;2AN;258;R;</p> <p>    ;;16.RELEASE CODE;"00";2;L;;2AN;260;R;</p> <p>    ;;18.END OF RECORD;</p> <p>.....</p> <p>    ; EMDEON CLAIM IEVEL STATUS TABLE: Contains the Descriptions for all data to be ;</p> <p>    ; gathered for E01 File. Used by \$TEXT to gather all data &amp; set up format ;</p> <p>    ; Each record is described in detail for the following Parameters: ;</p> <p>    ; This table is used as a block of data(a record) for each claim to be processed. ;</p> <p>    ; Therefore, separating this from the header and trailer tables allows multiple ;</p> <p>    ; calls to this table. ;</p> <p>.....</p> <p>    ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ;</p> <p>    ; FIELD NAME: EMDEON File FIELD DESCRIPTOR ;</p> <p>    ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ;</p> <p>    ; JUSTIFY FLAG: L=LEFT, R=RIGHT ;</p> <p>    ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ;</p> <p>    ; PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ;</p> <p>    ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION</p> <p>ONLY ;</p> <p>    ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ;</p> <p>.....</p> <p>    ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP) ;</p> <p>    ; 8/5/11 DLB FIELDS 12,14,15: TYPO FIXED FOR PROVIDER INFO (ADDED ^ TO CHMXCLB) ;</p> <p>    ; 8/5/11 DLB FIELD 41: CHANGED INDEX FROM ^CHMXCLE(I,100 TO ^CHMXCLE(I,0 ;</p> <p>.....</p> <p>EMDEONCLM ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA</p> <p>LENGTH/PATTERN";FIELD START LOCATION;FIELD USE</p>	

Routine Name	CHMXWB21
	<pre> ;;1.RECORD ID;"CLM";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.MAINTENANCE TYPE CODE;\$\$(\$G(RUNTYPE)="F":"021",\$G(RUNTYPE)="A":"021",\$G(RUNTYPE)="P":"021",\$G(RUNTY PE)="H":"030",1:"001");3;L;;3AN;19;R; ;;5.BILLING PROVIDER FEDERAL TAX ID;BPFTI;9;L;;9N;22;C; ;;6.BILLING PROVIDER PAYER NUMBER;BPPN;50;L;;50AN;31;C; ;;7.BILLING PROVIDER NATIONAL ID;BPNID;10;L;;10N;81;C; ;;8.BILLING PROVIDER LAST NAME;BPLNAME;60;L;;60AN;91;C; ;;9.BILLING PROVIDER FIRST NAME;BPFNAME;35;L;;35AN;151;O; ;;10.BILLING PROVIDER MIDDLE NAME;BPMNAME;25;L;;25AN;186;O; ;;11.BILLING PROVIDER NAME SUFFIX;BPNAMEX;10;L;;10AN;211;O; ;;12.SERVICE PROVIDER FEDERAL TAX ID;SPFTID;9;L;;9N;221;C; ;;13.SERVICE PROVIDER PAYER NUMBER;SPPN;50;L;;50AN;230;C; ;;14.SERVICE PROVIDER NATIONAL ID;SPNID;10;L;;10N;280;C; ;;15.SERVICE PROVIDER LAST NAME;SPLNAME;60;L;;60AN;290;C; ;;16.SERVICE PROVIDER FIRST NAME;SPFNAME;35;L;;35AN;350;O; ;;17.SERVICE PROVIDER MIDDLE NAME;SPMNAME;25;L;;25AN;385;O; ;;18.SERVICE PROVIDER NAME SUFFIX;SPNAMEX;10;L;;10AN;410;O; ;;19.EMPLOYER IDENTIFICATION NUMBER;EMPIDNUM;80;L;;80AN;420;C; ;;20.EMPLOYER NAME;EMPNAME;60;L;;60AN;500;C; ;;21.SUBSCRIBER ID;SUBSCID;80;L;;80AN;580;C; ;;22.SUBSCRIBER LAST NAME;SUBLNAME;60;L;;60AN;660;C; ;;23.SUBSCRIBER FIRST NAME;SUBFNAME;35;L;;35AN;720;O; ;;24.SUBSCRIBER MIDDLE NAME;SUBMNAME;25;L;;25AN;755;O; ;;25.SUBSCRIBER NAME SUFFIX;SUBNAMEX;10;L;;10AN;780;O; ;;26.PATIENT ID;PATID;80;L;;80AN;790;O; ;;27.PATIENT LAST NAME;PATLNAME;60;L;;60AN;870;R; ;;28.PATIENT FIRST NAME;PATFNAME;35;L;;35AN;930;R; ;;29.PATIENT MIDDLE NAME;PATMNAME;25;L;;25AN;965;O; ;;30.PATIENT NAME SUFFIX;PATNAMEX;10;L;;10AN;990;O; ;;31.PATIENT DATE OF BIRTH;PATDOB;8;L;;8DT;1000;R; ;;32.PATIENT GENDER;PATGENDR;1;L;;1AN;1008;O; ;;33.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;1009;O; ;;34.CLAIM CHARGE AMOUNT;CLMCHRG;18;L;;18N;1059;R; ;;35.CLAIM PYMT AMT;CLMPMT;18;R;;18N;1077;C; ;;36.CLAIM ADJ/PAYMENT DATE;CAPD;8;L;;8DT;1095;C; ;;37.CHECK/EFT DATE;CHKEFTDATE;8;L;;8DT;1103;O; ;;38.CHECK/EFT NUMBER;CHKEFTNUM;16;L;;16AN;1111;O; ;;39.BILL TYPE;BTYPE;3;L;;3AN;1127;O; ;;40.PAYER CLAIM ID NUMBER;PCIDNUM;50;L;;50AN;1130;R; ;;41.PATIENT ACCOUNT NUMBER;PATACT;50;L;;50AN;1180;O; ;;42.PHARMACY PRESCRIPTION NUMBER;PRSCNUM;50;L;;50AN;1230;O; ;;43.VOUCHER IDENTIFIER;VOUCHID;50;L;;50AN;1280;O; ;;44.APP/LOCATION SYSTEM ID;LOCSYSID;50;L;;50AN;1330;O; ;;45.GROUP NUMBER;GRPNUM;50;L;;50AN;1380;O; ;;46.CLAIM SERVICE DATE START;CLMSTDT;8;L;;8DT;1430;R; ;;47.CLAIM SERVICE DATE END;CLMENDT;8;L;;8DT;1438;R; ;;48.END OF RECORD;  ..... ; EMDEON CLAIM STATUS RECORD DETAIL: Contains the Descriptions for all data to be ; ; gathered for Claim level and Line Level records. Used by \$TEXT to gather all ; ; data &amp; set up format. ; </pre>

Routine Name	CHMXWB21
	<pre> ; Each record is described in detail for the following Parameters: ; ; NOTE: ; ; For each CLM record, there must be at least one STC record. ; ; For a single CLM record, The following is true about DTL records: ; ; 1) Payer can send any number of DTL records (including NONE) ; ; 3) Each DTL record sent corresponds to Line Level info for the CLM record. ; ; 4) For each DTL record sent, there must be at least one STC record. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----; ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTP) ; ;-----;  EMDEONSTC ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"STC";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;3;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;13;R; ;;4.PAYER CLAIM ID NUMBER;PCINUM;50;L;;50AN;18;R; ;;5.LINE ITEM CONTROL NUMBER;LICNUM;50;L;;50AN;68;O; ;;6.STATUS INFORMATION EFF. DATE;DATESTAMP;14;L;;14N;118;R; ;;7.CLAIM STATUS CATEGORY CODE;RJCODE;3;L;;3AN;126;R; ;;8.CLAIM STATUS CODE;RJSTATUS;3;L;;3AN;129;R; ;;9.ENTITY CODE;ENTITY;3;L;;3AN;132;O; ;;10.DATA IN ERROR;DATERR;264;L;;264AN;135;O; ;;11.EMDEON STATUS CODE;ESCODE;5;L;;5AN;399;O; ;;13.END OF RECORD;  ;-----; ; EMDEONTRLR: Common Trailer for EMDEON Status Files ; ; NOTE: a single trailer record is generated for each output file. ; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: 277 File FIELD DESCRIPTOR ; ; LENGTH: 277 FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;-----;  EMDEONTRLR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"TRLR";4;L;;4AN;0;R; ;;2.RECORD COUNT;COUNT;10;R;;10N;4;R; ;;3.END OF RECORD; </pre>

Routine Name	CHMXWB21
<pre> ..... ; EMDEON LINE LEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for CRF File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; NOTE: there may be multiple line items for each claim record. This table ; ; defines the record to be generated for each line item. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; .....  EMDEONLI ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"DTL";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.PAYER CLAIM ID NUMBER;PCID;50;L;;50AN;19;R; ;;5.LINE ITEM CONTROL NUMBER;LICN;50;L;;50AN;69;R; ;;6.SERVICE QUALIFIER ID;SQID;2;L;;2AN;119;R; ;;7.SERVICE IDENTIFICATION CODE;SICODE;48;L;;48AN;121;R; ;;8.PROCEDURE MODIFIER 1;PRCM1;2;L;;2AN;131;O; ;;9.PROCEDURE MODIFIER 2;PRCM2;2;L;;2AN;133;O; ;;10.PROCEDURE MODIFIER 3;PRCM3;2;L;;2AN;135;O; ;;11.PROCEDURE MODIFIER 4;PRCM4;2;L;;2AN;137;O; ;;12.LI CHARGE AMOUNT;LICHARGA;18;L;;18N;139;R; ;;13.LINE ITEM PROV. PAYMENT AMOUNT;LIPPA;18;L;;18N;157;R; ;;14.REVENUE CODE;RVNUCODE;48;L;;48AN;175;C; ;;15.QUANTITY(UNITS OF SERVICE);QTYUOS;15;L;;15N;223;O; ;;16.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;238;O; ;;17.SRVC START DATE;SVCSTDATE;8;L;;8N;288;R; ;;18.SRVC END DATE;SVCENDATE;8;L;;8N;306;R; ;;20.END OF RECORD; </pre>	

### 3.2.2.2 Archive All Final Claim Status Files for Secondary Claims

#### Functional Requirement: 2.8.2.2 Save Final Claim Status Files – Secondary Claims

The Archiving of the CHAMPVA Final Claim Status files after transmittal to HCCH will follow the CP&E systems currently implemented methodology for the archiving of files sent to the HCCH. The system will use the CP&E existing methodology to perform the Archive process. The following routine places the file(s) in the proper VMS directory for pickup by the FTP MOVEit process.



Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.2	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		
<pre> GETFILE()   N DIRPATH   S EICNT=0,FILECNT=FILECNT+1   <b>S DIRPATH="CHAMPVA EDI:[KERMIT.WEBMD]"</b>   HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP   D NOW^%DTC   S FMDATE=%   S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")   S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")   S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND   <b>I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT"</b>   <b>E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT"</b>   ; CHMEDCOB SET IN CHMXWT1   E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME   I RUNTYPE="H" S DIRFILE=DIRFILE_"_H"   I RUNTYPE="O" S DIRFILE=DIRFILE_"_O"   S FLAG=\$\$OFILE^CHMXWBUT(DIRFILE,"NWS")   I 'FLAG D VMSERR Q ""   S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q </pre>		

### 3.2.2.3 Create Final Claim Status File Data for Secondary Claims

#### Functional Requirement: 2.8.2.3 Final Claim Status File Data – Secondary Claims

The CP&E system shall create the Final Claim Status File(s) based on the Emdeon / Hosted Real-Time Specification document and adhere to all file specifications and limitations. The specified file size limit of 500 mb per file will be modified to improve submission performance. The new file size limit will be 150 mb. The process will populate the following data in the Final Claim Status file for electronic secondary CHAMPVA claims when available:

- Header: Record ID = HDR - Required
- Header: File Group ID - Required
- Header: File Group Sequence Number - Required
- Header: File Group Count - Required
- Header: Creation Date - Required
- Header: Creation Time - Required
- Header: Trading Partner ID = VAFNH - Required
- Header: Submitter Name = VA Health Administration Center - Required
- Header: Payer Contact Name = CHAMPVA Customer Service - Required
- Header: Payer Support Telephone Number = 1-800-733-8387 (CHAMPVA) - Required
- Header: Payer Support Email Address - Optional
- Header: Load Type = I - Required
- Header: Payer Unique File Identifier - Required
- Header: File Type = CStat - Required
- Header: Version Code = 03 - Required
- Header: Release Code = 00 - Required
- Header: Record Terminator = CRLF - Required
- Claim Level Record Details: Record ID = CLM - Required
- Claim Level Record Details: Record Number = 2 and increments by 1 - Required
- Claim Level Record Details: Payer ID = VAHAC - Required
- Claim Level Record Details: Maintenance Type Code = 021 - Required
- Claim Level Record Details: Billing Provider Federal Tax ID - Conditional
  - Required when the Billing Provider Payer Assigned Number and the Billing Provider National Provider ID are not present
- Claim Level Record Details: Billing Provider Payer Assigned Number - Conditional
  - Required when the Billing Provider Federal Tax ID and the Billing Provider National Provider ID are not present
- Claim Level Record Details: Billing Provider National Provider ID - Conditional
  - Required when the Billing Provider Federal Tax ID and the Billing Provider Payer Assigned Number are not present

- Claim Level Record Details: Billing Provider Last Name/Organization Name - Optional
- Claim Level Record Details: Billing Provider First Name - Optional
- Claim Level Record Details: Billing Provider Middle Name - Optional
- Claim Level Record Details: Billing Provider Name Suffix - Optional
- Claim Level Record Details: Service Provider Federal Tax ID - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID is required, else not used
- Claim Level Record Details: Service Provider Payer Assigned Number - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID is required, else not used
- Claim Level Record Details: Service Provider National Provider ID - Conditional
  - If the Service Provider ID is different from the Billing Provider ID, either the Federal Tax ID, the Service Provider Payer Assigned Number, or the Service Provider National Provider ID is required, else not used
- Claim Level Record Details: Service Provider Last Name/Organization Name - Conditional
  - Only used if either Service Provider Federal Tax ID Number, Service Provider Payer Assigned Number, or the Service Provider National Provider ID is present
- Claim Level Record Details: Service Provider First Name - Optional
- Claim Level Record Details: Service Provider Middle Name - Optional
- Claim Level Record Details: Service Provider Name Suffix - Optional
- Claim Level Record Details: Employer Identification Number - Conditional
  - Required if Employer Name is present
- Claim Level Record Details: Employer Name - Conditional
  - Used in cases of Employer-Subscriber
- Claim Level Record Details: Subscriber ID - Conditional
  - Required when Subscriber Last Name is present
- Claim Level Record Details: Subscriber Last Name - Conditional
  - Used when the Subscriber is an insured individual
- Claim Level Record Details: Subscriber First Name - Optional
- Claim Level Record Details: Subscriber Middle Name - Optional
- Claim Level Record Details: Subscriber Name Suffix - Optional
- Claim Level Record Details: Patient ID - Optional
- Claim Level Record Details: Patient Last Name - Required
- Claim Level Record Details: Patient First Name - Required
- Claim Level Record Details: Patient Middle Name - Optional
- Claim Level Record Details: Patient Name Suffix - Optional
- Claim Level Record Details: Patient Date of Birth - Required

- Claim Level Record Details: Patient Gender - Optional
- Claim Level Record Details: HCCH Claim Number - Required
- Claim Level Record Details: Claim Charge Amount - Required
- Claim Level Record Details: Claim Payment Amount - Conditional
  - Required when Claim Status Category Code 1-3 = F1
- Claim Level Record Details: Claim Adjudication/Payment Date - Conditional
  - Required when first position of Claim Status Category Code = F
- Claim Level Record Details: Check/EFT Date - Optional
- Claim Level Record Details: Check/EFT Number - Optional
- Claim Level Record Details: Bill Type - Optional
- Claim Level Record Details: Payer Claim Identification Number - Required
- Claim Level Record Details: Patient Account Number - Required when provider claim number received
- Claim Level Record Details: Pharmacy Prescription Number - Optional
- Claim Level Record Details: Voucher Identifier - Optional
- Claim Level Record Details: Application or Location System Identifier - Optional
- Claim Level Record Details: Group Number - Optional
- Claim Level Record Details: Claim Service Date Start - Required
- Claim Level Record Details: Claim Service Date End - Required
- Claim Level Record Details: Record Terminator = CRLF - Required
- Claim Line Level Record Details: Record ID = DTL - Required
- Claim Line Level Record Details: Record Number - Required
- Claim Line Level Record Details: Payer ID - Required
- Claim Line Level Record Details: Payer Claim Identification Number - Required
- Claim Line Level Record Details: Line Item Control Number - Required
- Claim Line Level Record Details: Service Qualifier ID - Required
- Claim Line Level Record Details: Service Identification Code - Required
- Claim Line Level Record Details: Procedure Modifier 1 - Optional
- Claim Line Level Record Details: Procedure Modifier 2 - Optional
- Claim Line Level Record Details: Procedure Modifier 3 - Optional
- Claim Line Level Record Details: Procedure Modifier 4 - Optional
- Claim Line Level Record Details: Line Item Charge Amount - Required
- Claim Line Level Record Details: Line Item Provider Payment Amount - Required
- Claim Line Level Record Details: Revenue Code - Optional
- Claim Line Level Record Details: Quantity (Units of Service) - Optional
- Claim Line Level Record Details: HCCH Claim Number - Required
- Claim Line Level Record Details: Date of Service Start - Required
- Claim Line Level Record Details: Date of Service End - Required
- Claim Line Level Record Details: Record Terminator = CRLF - Required
- Claim Status Record Detail: Record ID = STC - Required

- Claim Status Record Detail: Record Number - Required
- Claim Status Record Detail: Payer ID - Required
- Claim Status Record Detail: Payer Claim Identification Number - Required
- Claim Status Record Detail: Line Item Control Number - Required
- Claim Status Record Detail: Status Information Effective Date - Required
- Claim Status Record Detail: Claim Status Category Code - Required
- Claim Status Record Detail: Claim Status Code - Required
- Claim Status Record Detail: Entity Code - Optional
- Claim Status Record Detail: Data in Error - Optional
- Claim Status Record Detail: HCCH Status Code - Optional
- Claim Status Record Detail: Record Terminator = CRLF - Required
- Trailer Record Details: Record ID = TRLR - Required
- Trailer Record Details: Record Count - Required
- Trailer Record Details: Record Terminator = CRLF – Required

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.3	
Related Options		
<b>Related Routines</b>	<b>Routines “Called By”</b>	<b>Routines “Called”</b>
	<b>Taskman process</b>	<b>CHMXWBUT</b>
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		

Routine Name	CHMXWB21
	<pre> CHMXWB21 ;HAC/DLB;WEB 5010 277 UNSOLICITED STATUS/DRIVER PART 1;10/25/10 ;;1.0;CHAMPVA SYSTEM;;JULY 11,2011;Build 1 ;;V1.0; ;;CALLED BY ^CHMXG001 TO PERFORM THE "ACK" STATUS ;;CALLED BY TASKMAN TO PERFORM THE "PENDING" AND "FINAL" STATUS ;;CALLS ENTRY POINTS TO CHMXWB22,CHMXWBUT ;;HR-COB-Medicare-A/B-Begin-9372 (27-May-2010) ** ** ** 10/18/2011 ADDED TSTBLDCLM() ENTRY POINT TO VALIDATE THE REJECT VALUES BEING ** REPORTED IN THE CSTAT FILE ** 10/20/2011 MODIFIED THE LINE() FUNCTION TO FIX INCORRECT REJECT STATUS ** REPORTING IN THE CSTAT FILE ** 10/21/2011 ADDED THE SIX DATE GATHERING FUNCTIONS TO SUPPORT THE RULE ** PROVIDED BY BUSINESS GROUP FOR I/P/D CLAIMS ** 10/21/2011 ADDED THE GETCLMTYP() FUNCTION TO DETERMINE THE I/P/D CLAIM TYPE ** FOR BOTH 4010 AND 5010 INCOMING CLAIMS ** 10/24/2011 CHANGED THE DATERANGE FUNCTION TO A EXISTING DATE FUNCTION ** IN THE UTILITY ROUTINE: CHMXWBUT.INT ** 10/25/2011 CHANGED THE RENDERING PHYSICIAN NPI RETRIEVAL TO GET THE DATA ** FROM THE CHMXCLE(I,64) TIN DATA FIELD. ** 10/25/2011 ADDED A TEST FOR THE "U" VALUE RECEIVED FROM EMDEON TO RETURN "" ** IF NOT A "M" OR "F" (ONLY ACCEPTED VALUES) ** 10/26/2011 REMOVED ERROR MESSAGE OUTPUT FOR ERROR TRAP HANDLER ** ** ** 10/26/2011 CHANGED THE INSTITUTIONAL AND DENTAL START/END DATES FOR CLAIM ** LEVEL REPORTING TO GET START FROM ^CHMXCLE(I,1),FIELD 1 AND END ** TO ^CHMXCLE(I,1) FIELD 2 ** 11/4/2011 CHANGED THE DATE CHECK ROUTINE GETCLMI() TO RETURN THE CORRECT ** DATE FOR THE DESIRED FLAG ** ** ;1.0 IS THE INITIAL VERSION (HIPAA Ready LLC) ;2.0 UPDATES TO THE STATUS REPORT PER HIPAA II SPECIFICATIONS. ; *** Filenames should not be hard coded in files. See ^CHMX277 TRACKING ; GLOBAL FILE for the destination directory/filenames for the 5010 status ; files. Q  ***** ; PRODUCTION ENTRY POINTS TO PROCESS THE CURRENT STATUS. ; *****  EPACK(CHMXI) ; SET UP THE FILE/BATCH INDEX (PROVIDED BY CHMXG001) ; CHMXI I INDEX INTO ^CHMXCL() N COUNT,DIRFILE,GRPCNT,FILECNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR,CNT N MAXCNT,EICNT,CHA,CHMEDCOB,TPINT S FILECNT=0,RUNTYPE="A",GRPCNT=1,EICNT=0 S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER D INIT S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" S CHTPI=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:CHTPI="" ; GET CLAIM T/P FOR FILE BEING CHECKED S CHTPI=\$O(^CHMXTP("C",CHTPI,"")) Q:CHTPI="" D ; GET "LOCAL" TP ID S IDX=\$O(^CHMX277("B",CHTPI,0)) </pre>

Routine Name	CHMXWB21
<pre> S:\$P(^CHMX277(IDX,0),"^",4) ["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER I CHMEDCOB=0 S TPINT=\$P(^CHMX277(IDX,0),"^") I TPINT="" S CHMEDCOB=+\$P(\$G(^CHMXTP(TPINT,0)),"^",11) D GETCHCLEI(CHMXI,CHTPI) ; RETRIEVE ^CHMXCLE INDEX Q: ^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPPENDTST N TEST S TEST=1 D EPPEND Q  EPPEND N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,CHTPI,CHTPARR N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA,MAXCNT,CHMEDCOB,IDX ; FILEMAN UPDATE VARIABLES N CHTO,CHFROM,TRKI,EICNT S RUNTYPE="P",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF .S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER .S IDX=\$O(^CHMX277("B",CHTPI,0)) .S:\$P(^CHMX277(IDX,0),"^",4) ["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .K CHTPARR .S (CHFROM,CHTO,FILECNT)=0 ; KILL TRADING PARTNER ARRAY .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX .Q:(\$P(\$G(^CHMX277(TRKI,0)),"^",3)=0) .S CHTPARR(CHTPI)="" ; SET CHTPARR UP FOR GETPENDEI .D GETTOFROM(CHTPI) Q:(CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETPENDEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q: ^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) ; UPDATE TRACKING GLOBAL VIA FILEMAN FUNCTION Q  EPFINAL ; entry point to start Final Status Creation file N RUNTYPE,GRPCNT,EICNT,GROUPID,CHTPI,CHTPARR,CHFROM,CHTO,FILECNT N TRKI N COUNT,DIRFILE,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,FMDATE N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA,MAXCNT ; FILEMAN UPDATE VARIABLES S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF </pre>	

Routine Name	CHMXWB21
<pre> .K CHTPARR S (CHFROM,CHTO,FILECNT,CHMEDCOB)=0 ; KILL TRADING PARTNER ARRAY; ASSUME NOT A "COB" TRADING PARTNER .S TRKI=\$O(^CHMX277("B",CHTPI,0)) ; GET TRACKING GLOBAL INDEX .I \$P(\$G(^CHMX277(TRKI,0)),",",3)=0 Q .I \$P(^CHMX277(TRKI,0),",",4)["COB" S CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .S CHTPARR(CHTPI)=" .D GETTOFROM(CHTPI) Q:(('CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETFINALEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q: ^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  ***** ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; *****  EPREPTST N TEST S TEST=1 D EPREPLACEMENT Q  EPREPLACEMENT N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT,CHMEDCOB,IDX,CHTPX S RUNTYPE="H",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="REPLACEMENT" S TP=\$\$LISTTP^CHMXWB22 ; SELECT TRADING PARTNER S CHMEDCOB=0 ; ASSUME NOT A "COB" TRADING PARTNER S IDX=\$O(^CHMX277("B",TP,0)) S:\$P(^CHMX277(IDX,0),",",4)["COB" CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$O(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHXI="" .F S CHXI=\$O(^CHMXCL("B",IDATE,CHXI)) Q:CHXI="" D ..S CHMXI=CHXI ..I \$D(TEST) W !,"CHECKING:",\$\$DTCVRT(\$E(IDATE,1,7)) ; SHOW FILE DATES BEING PROCESSED ..S CHA=\$O(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S CHTPX=\$P(\$G(^CHMXCLA(CHX,1)),",",1) Q:CHTPX="" ; GET CLAIM T/P FOR FILE BEING CHECKED ..S CHTPX=\$O(^CHMXTP("C",CHTPX,"")) ; CHECK SELECTED VS CLAIM TP ID ..I TP=CHTPX D ; FILE MATCHES THE SELECTED TP ...S CHTPI=TP ; CHTPI IS USED DOWNSTREAM ...I \$D(TEST) W " PROCESSING:",\$\$DTCVRT(\$E(IDATE,1,7)) ...I \$D(TEST) W !," EPREPLACEMENT: CHMXI= ",CHMXI," CHA= ",CHA," CHTPI= ",CHTPI," </pre>	



Routine Name	CHMXWB21
<pre> TP= ",TP ...S CHTPARR(TP)=" D GETCHCLEI(CHMXI,CHTPI) ; IF TP VALID, PROCESS FILE I \$D(TEST) W !,"REPLACE: \$J= ",\$J," REC COUNT= ",\$G(^ZSC(\$J,0))," CHTPI= ",CHTPI," TP= ",TP Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  DTCVRT(DATE) ; CONVERT DATE TO EXTERNAL FORMAT N EXTDATE S EXTDATE=\$E(DATE,4,5)_"-"\$E(DATE,6,7)_"-"\$E(DATE,1,3)+1700 Q EXTDATE  ..... ; HISTORICAL STATUS RECORD GENERATION ENTRY POINTS ; .....  EPOVERTST N TEST S TEST=1 D EPOVERLAY Q  EPOVERLAY N COUNT,DIRFILE,GRPCNT,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,RUNTYPE,FMDA TE,MAXCNT N POP,CHMXI,CHENDI,FROM,TO,TP,RUNTYPE,CHTPI,EICNT S RUNTYPE="O",FILECNT=0,GRPCNT=1,EICNT=0,CHTPI=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S GROUPID="OVERLAY" S TP=\$\$LISTTP^CHMXWB22 D GETDTE^CHMXWBUT(.FROM,.TO) ; GET DATE RANGE VALUES S IDATE=FROM,EDATE=TO_".999999" F S IDATE=\$Q(^CHMXCL("B",IDATE)) Q:IDATE&gt;EDATE Q:IDATE="" D .S CHMXI="" .F S CHMXI=\$Q(^CHMXCL("B",IDATE,CHMXI)) Q:CHMXI="" D ..S CHA=\$Q(^CHMXCLA("B",CHMXI,"")) Q:CHA="" ..S TPTIN=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) Q:TPTIN="" ..I TP=\$Q(^CHMXTP("C",TPTIN,"")) S CHTPARR(TP)=" D GETCHCLEI(CHMXI,TP) Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE D CREATEFILE Q  ..... ; INIT PERFORMS LOCAL VARIABLE INITIALIZATION REQUIREMENTS FOR THE STATUS ; ; PROCESSES. ; .....  INIT ; K ^ZSC(\$J) S ^ZSC(\$J,0)=0 K TMP S GROUPID="",CHTPI=0 I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC </pre>	

Routine Name	CHMXWB21
<pre> .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND Q  ; ; EPSTAT IS THE ENTRY POINT FOR THE ON-DEMAND REPORT FOR STATISTICS ; ;  EPSTAT D ONDEMAND^CHMXWB22 Q  EPREGEN D REGEN^CHMXWB22 Q  ; ; FILE CREATION FUNCTIONS ; ;  CREATEFILE ; S MAXCNT=500000 ; HAPE POR 11/21/13 force max file size under 150 meg I RUNTYPE="F" S MAXCNT=175000 D FILELOOP Q  FILELOOP ; BUILD CLAIMS FROM ^ZSC(\$J) ARRAY (^CHMXCLE(I) VALUES) N CHCLEI,PDI D GETGRPCNT ; GET ESTIMATED GROUP FILE COUNT Q:GRPCNT=0 D GETFILE U DIRFILE ; CREATE ORIGINAL FILE D BLDHDR ; BLDHDR SETS COUNT=2 (SEE EMDEON SPEC) S CHCLEI=0 F S CHCLEI=\$O(^ZSC(\$J,CHTPI,CHCLEI)) Q:CHCLEI="" D .S EICNT=EICNT+1 .S STVAL=\$G(^ZSC(\$J,CHTPI,CHCLEI)) .D BLDCLM(CHCLEI,STVAL) ; GENERATE CLAIM/STC,DTL/STC RECORDS .D CHKSEQ ; CHECK FOR 500MB LIMIT, CREATE NEW FILE D BLDTRL ; BUILD THE TRAILER RECORD D CLOSEFILE ; CLOSE CURRENT FILE Q  GETGRPCNT ; ESTIMATE THE TOTAL NUMBER OF FILES REQUIRED N RECCNT,X S RECCNT=0 S X="" F S X=\$O(^ZSC(\$J,CHTPI,X)) Q:X="" S RECCNT=RECCNT+1 S GRPCNT=RECCNT\MAXCNT ; INTEGER DIVISION OF RECORD COUNT S:RECCNT#MAXCNT&gt;0 GRPCNT=GRPCNT+1 ;W !,"GRPCNT=",GRPCNT </pre>	

Routine Name	CHMXWB21
<pre> Q CHKSEQ      ; HANDLE THE FILE SEQUENCE NUMBERING REQUIREMENTS I EICNT&gt;MAXCNT D .D BLDTRL      ; OUTPUT TRAILER RECORD .D CLOSEFILE .D GETFILE U DIRFILE .D BLDHDR      ; BLDHDR RESETS "COUNT" VARIABLE Q  FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN Q:"APF"[RUNTYPE      ; A=ACK, P=PENDING, F=FINAL N CHNODEDEF,BREF,CHIKVAL,DRDATA,TRKDATA S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^.03="_DIRFILE_".04="_COUN T_".05="_CHMXI S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20,"TRKDATA=".02=1^.03="_DIRFILE_".04="_CHTO _".05="_CHFROM_".06="_COUNT S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30,"TRKDATA=".02=1^.03="_DIRFILE_".04="_CHTO _".05="_CHFROM_".06="_COUNT S BREF=FMDATE S CHIKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT D ADD^CHHRLIBFM(CHNODEDEF,CHIKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL I RUNTYPE="F" N X,X1 F S X=\$O(^ZSC1(\$J,X)) Q:X="" F S X1=\$O(^ZSC1(\$J,X,X1)) Q:X1="" I \$D(^CHMPAY("E",X,X1)) S ^CHMPAY("E",X,X1)=1 Q  ..... ;TEST ENTRY POINT TO TEST THE BLDCLM FUNCTION FROM A PDI VALUE ; THERE ARE MULTIPLE CROSS REFERENCES (SOME DOCUMENTED, SOME UNDOCUMENTED ; THAT ARE UTILIZED IN THE CLAIM PROCESSING SYSTEM. THE CVTPDI AND TSTBLDCLM ; FUNCTIONS UTILIZE THE MAJORITY OF THESE CROSS REFERENCES. .....  CVTPDI(CHREFN) N CHPCN,CHMXCLI,CHMXID,CHCLFI,SDATE,EDATE,CHCLFI,CHCLEI,IDX I '\$D(^CHMXCLE("PDI",CHREFN)) W !,"NO ^CHMXCLE("PDI",",CHREFN,") XREF" Q S CHPCN="" S1 S CHPCN=\$O(^CHMXCLE("PDI",CHREFN,CHPCN)) I CHPCN="" Q ;VALID PDI,NO DATA S CHMXCLI=0 S2 S CHMXCLI=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI)) I 'CHMXCLI Q ;VALID PDI, NO DATA S CHMXID="" S3 S CHMXID=\$O(^CHMXCLE("PDI",CHREFN,CHPCN,CHMXCLI,CHMXID)) I CHMXID="" Q S CHCLEI=\$P(CHMXID,"*",4) W !,"CLM SDATE [^CHMXCLE(I,1),1]=",\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) W !,"CLM EDATE [^CHMXCLE(I,1),2]=",\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) S CHCLFI=0 W !,"LINE ITEM DATE VALUES" F IDX=1:1 S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:'CHCLFI D ; CHECK ALL LINE ITEM DATES .W !,IDX,". ",", " L/I SDATE [^CHMXCLF(I,1),11]=",\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) </pre>	

Routine Name	CHMXWB21
<pre> .W !," L/I END [^CHMXCLF(I,1)),12)]=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) .W !," THERAPY SDATE [^CHMXCLF(I,2.5)),1)]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",1) .W !," LAST CERT DATE [^CHMXCLF(I,2.5)),2)]=\$P(\$G(^CHMXCLF(CHCLFI,2.5)),^",2) Q  TSTBLDCLM(CHPDI)   N CHPCN,CHCLI,CHCLAI,CHCLEI,STVAL,COUNT,PDI,BUFI,RUNTYPE,CLMTYPE,TEST   S (STVAL,CHPCN)="",(COUNT,CHCLEI,CHCLAI,PDI)=0,TEST=1   S CHPCN=0,CHPCN=\$O(^CHMXCLE("PDI",CHPDI,CHPCN))   I CHPCN="" W !,"NO CHPCN VALUE" Q   E W !,"PATIENT CONTROL # [^CHMXCLE("PDI",CHPDI,PCN)]= ",CHPCN      ; PATIENT CONTROL NUMBER   S CHCLI=0,CHCLI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI))   I CHCLI="" W !,"NO CHCLI VALUE" Q   S STVAL=0,STVAL=\$O(^CHMXCLE("A",CHCLI,STVAL)) W !,"XREF CLAIM STATUS= ",STVAL   S BUFI="",BUFI=\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))   I BUFI="" W !,"NO BUFFER INDEX VALUES" Q   W !,"XREF [\$O(^CHMXCLE("PDI",CHPDI,CHPCN,CHCLI,BUFI))] BUFFER INDICES= ",BUFI   S CHCLEI=\$P(BUFI,"*",4)   I CHCLEI="" W !,"NO CHCLEI VALUE RECOVERED"   S RUNTYPE="H"   S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; RETRIEVE PDI   S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4)   W !,"PDI VALUE FROM [^CHMXCLE(I,100)),2 OR 4]= ",PDI   S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),^",1) ;GET "I" VALUES FOR ALL BUFFER FILES   S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),^",1)   S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),^",1)   S CHCLI=\$P(^CHMXCLA(CHCLAI,0),^",1)   W !!,"X12 837 BUFFER [CHMXCL(I)= ",?30,CHCLI   W !,"TRANSACTION [CHMXCLA(I)]= ",?30,CHCLAI   W !,"PROVIDER BUFFER [CHMXCLB(I)]= ",?30,CHCLBI   W !,"PATIENT BUFFER [CHMXCLC(I)= ",?30,CHCLCI   W !,"CLM BUFFER [CHMXCLE(I)= ",?30,CHCLEI   W !,"L/I BUFFER [CHMXCLF(I)]= ",?30,\$O(^CHMXCLF("B",CHCLEI,0))   S CLMTYPE=\$\$GETCLMTYP   W !,"VERSION-FUNCTIONAL TYPE [^CHMXCLA(I,0)),13)]="   ",\$P(\$G(^CHMXCLA(CHCLAI,0)),^",13)   W !!,"CLAIM TYPE="   ",\$\$S(CLMTYPE="I":"INSTITUTIONAL",CLMTYPE="P":"PROFESSIONAL",CLMTYPE="D":"DENTAL") ; I/P/D CLAIM   W !,"STATUS (ACK/PENDING)=\$,\$S(\$D(^CHMPAY("C",PDI))=0:"ACK",1:"PENDING")   D CVTPDI(CHPDI)   W !!   D BLDCLM(CHCLEI,"")   Q  ..... ; BLDCLM: ENTRY POINT TO BUILD THE CSTAT RECORDS ; THIS FUNCTION PROVIDES AN EXTERNAL ENTRY POINT TO GENERATE THE ; EMDEON CLAIM,CLAIM STATUS,DETAIL,AND DETAIL STATUS RECORDS ..... </pre>	
<pre> <b>BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS</b> <b>; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE</b> <b>; STVAL STATUS VALUE AFTER FROMNT END EDITS</b> </pre>	

Routine Name	CHMXWB21
	<pre> N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ; TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ; GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0,"^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) </pre>

Routine Name	CHMXWB21
<pre> I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D . S CHMPAY=0,OK=1 . F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=0 Q .. S OK=0 .. Q Q OK  ..... ; FUNCTIONS TO GATHER INFORMATION REGARDING REJECT REASONS ; .....  BLDACK ; POPULATE CHRJARR REJECT REASON/REJECT CODE VALUES K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY S REJCODES="" D GTRXRJRSN^CHMXWBUT(CHCLAI) ; TRANSACTION "101" NODE REJECT REASONS D GPRORJRSN^CHMXWBUT(CHCLBI) ; PROVIDER "101" NODE REJECT REASONS D GPATRJRSN^CHMXWBUT(CHCLCI) ; PATIENT "101" NODE REJECT REASONS D GCLMRJRSN^CHMXWBUT(CHCLEI) ; CLAIM "101" NODE REJECT REASONS I CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" ;I \$D(TEST) W !,"BLDACK: RJCNT=",CHRJARR(0)," RJARR(1,1)=",CHRJARR(1,1) Q  ..... ; RECORD BUILDING FUNCTIONS: \$TEXT FILES PROVIDE FIELD DEFINITIONS ; .....  BLDHDR ;BUILD HEADER RECORD N LN,REC,STR,LOADTYPE S (STR,LN,REC)="",COUNT=2 ; EMDEON SPEC: REC. COUNT STARTS @ 2 D GETHDRVAL ; SET UP HEADER VARIABLES F LN=1:1 S STR=\$T(EMDEONHDR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="" Q  CLM ; N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDY,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING </pre>	

Routine Name	CHMXWB21
	<pre> S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q </pre>
<pre> LINE ; BUILD DTL RECORDS FOR LINE ITEMS N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDATE,SVCENDATE N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D .D GDTLVARS .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),^") .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65*" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1*" Q ... S CHRJARR(1,1)="F0*1*" ... Q .. Q .D BLDSTC("DTL") Q </pre>	<pre> </pre>
<pre> BLDSTC(TYPE) ; WRITE OUT THE STC RECORDS FROM CHRJARR REJECT ARRAY ; TYPE CLAIM OR DETAIL LEVEL EMDEON STATUS VALUES &amp; LINE ITEM CONTROL NUMBER N PCINUM,LICNUM,DATERR,ESCODE N IDX,JDX,REJCODES,ECNT N LN,STR,REC S REJCODES=0 S ECNT=CHRJARR(0) ; FRONT END EDIT REJECT COUNT F IDX=1:1 Q:(IDX&gt;ECNT) D ; PROVIDE STC RECORD FOR EACH ERROR .F JDX=1:1 Q:\$G(CHRJARR(IDX,JDX))="" D ; GET EACH REJECT VALUE STORED ..S (RJCODE,RJSTATUS,ENTITY)="" ..D GETSTCVAL(TYPE) ; EMDEON STATUS RECORD VALUES FOR CLM OR DTL ..S REJCODES=\$G(CHRJARR(IDX,JDX)) ..;W:\$D(TEST) !,"BLDSTC: REJCODE STR=",REJCODES,! ; DEBUG: ONLY OUTPUTS IF "TEST" IS DEFINED ..S RJCODE=\$P(REJCODES,"*",1),RJSTATUS=\$P(REJCODES,"*",2),ENTITY=\$P(REJCODE S,"*",3) </pre>	<pre> </pre>

Routine Name	CHMXWB21
<pre> ..S REC="" ..F LN=1:1 S STR=\$T(EMDEONSTC+LN) Q:STR["END OF RECORD" D ; GENERATE THE STATUS (STC) RECORD ...I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ...E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ..W REC,! S REC="" S COUNT=COUNT+1 Q  BLDTRL ; TRAILER RECORD N LN,REC,STR S COUNT=COUNT-2 ; ADJUST COUNT FOR START OF 2 (SEE EMDEON SPEC) S (STR,LN,REC)=" F LN=1:1 S STR=\$T(EMDEONTRLR+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) ; 10/20/2010 ADDED "ELSE" W REC,! S REC="" Q  ..... ; GENERIC FUNCTIONS SUPPORTING THE RECORD BUILDING ; .....  GETFILE() N DIRPATH S EICNT=0,FILECNT=FILECNT+1 S DIRPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" HANG 1 ; POSITIVELY GUARANTEE NO TWO FILES HAVE SAME TIMESTAMP D NOW^%DTC S FMDATE=% S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND I \$D(TEST) S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_TEST.CSTAT" E I CHMEDCOB S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC_COB.CSTAT" ; CHMEDCOB SET IN CHMXWT1 E S DIRFILE=DIRPATH_DATESTAMP_"_VAHAC.CSTAT" ; PROD FILENAME I RUNTYPE="H" S DIRFILE=DIRFILE_"_H" I RUNTYPE="O" S DIRFILE=DIRFILE_"_O" S FLAG=\$\$OFIL^CHMXWBUT(DIRFILE,"NWS") I 'FLAG D VMSERR Q "" S \$P(^CHMXCL(CHMXI,80),"^",5)=DIRFILE ; RECORD THE CSTAT FILENAME ;E W !,"OPENED ",DIRFILE,! Q  TST N FMDATE,TIME S FMDATE=3110901.1234 W !,FMDATE S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") W !,FMDATE S TIME=\$E(FMDATE,9,14) W !,TIME S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") </pre>	



Routine Name	CHMXWB21
<pre> W !,TIME Q  CLOSEFILE C DIRFILE Q  GETCHCLEI(CHCLI,CHTPI) ; BUILD ARRAY OF CLAIM INDICES (CHMXCLE(I) VALUES) ; CHCLI I INDEX FOR ^CHMXCL() FILE (CLAIM FILE RECEIVED) N STAT,CHCLAI,PDI,CHIDXS,CHCLEI,RECCNT,CTPI S CTPI=0 S STAT="",STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) I \$D(TEST) W:STAT="" !,"NULL STAT" Q:STAT="" ; RETRIEVE STATUS VALUE S CHCLAI="",CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) I \$D(TEST) W:CHCLAI="" !,"NULL CHCLAI" Q:CHCLAI="" ; RETRIEVE THE ^CHMXCLA(I) VALUE I \$D(TEST) W !," GETCHCLEI: READING:",CHCLI," STATUS= ",STAT," ^CHMXCLA(I)= ",CHCLAI," TP= ",CTPI S STAT="" F S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) Q:STAT="" D .S CHCLAI="" .F S CHCLAI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI)) Q:CHCLAI="" D ..S PDI=0 ..F S PDI=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI)) Q:PDI D ...S CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,0)) ...Q:CHIDXS="" ; PDI  RETRIEVES BUFFER IDXs ...S CHCLEI=\$P(CHIDXS,"*",3) ...S ^ZSC(\$J,CHTPI,CHCLEI)=STAT,RECCNT=\$I(^ZSC(\$J,0)) Q  GETTP(CHA) N TPID,ID S ID="" S TPID=\$P(\$G(^CHMXCLA(CHA,1)),"^",1) S:TPID ID=\$O(^CHMXTP("C",TPID,"")) Q IDER  GETPENDEI(CHFROM,CHTO) ; GET CHCLEI FROM PENDING ^CHMPAY(I) FROM/TO VALUES K ^ZSC(\$J) N IDX,CHJK,PDI,CHCLEI,CHCLAI,CTPI,RECCNT S RECCNT=0 F IDX=CHFROM:1 Q:IDX&gt;CHTO D .S CHJK=\$O(^CHMPAY(IDX,"PDI",0)) Q:CHJK="" .S PDI=\$P(^CHMPAY(IDX,"PDI",CHJK,0),"^",1) Q:PDI="" .S CHCLEI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",6) Q:CHCLEI="" .S CHCLAI=\$P(\$G(^CHMIMAGE(PDI,"BUFF")), "^",3) Q:CHCLAI="" .S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM .I CHCLEI I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1 S ^ZSC(\$J,0)=RECCNT Q  <b>GETFINALEI(CHFROM,CHTO) ; GET CHCLEI FOR FINAL CLAIM STATUS</b> <b>K ^ZSC(\$J),^ZSC1(\$J)</b> <b>N CHCLEI,CHCLAI,CTPI,RECCNT,CLC,CLB,CHPIDX,CHPPDI</b> <b>S RECCNT=0</b> <b>S %H=\$H-180 D YMD^%DTC S CHPPDATE=X</b> <b>F S CHPPDATE=\$O(^CHMPAY("E",CHPPDATE))</b> </pre>	

Routine Name	CHMXWB21
<pre> Q:CHPDATE="" S CHPIDX="" F S CHPIDX=\$O(^CHMPAY("E",CHPDATE,CHPIDX)) Q:CHPIDX="" I \$G(^CHPIDX))'=1 D   . S CHPPDI=\$O(^CHMPAY(CHPIDX,"PDI",0)) Q:CHPPDI=""   . I CHPPDI=\$P(\$G(^CHMPAY(CHPIDX,"PDI",PDI,0)),")"   . I CHPPDI="" Q   . S CHCLEI=\$O(^CHMXCLE("D",CHPPDI,"")) I CHCLEI="" Q   . I \$\$FINAL(CHCLEI) D   .. S CLC=\$P(^CHMXCLE(CLE,0),")"   .. S CLB=\$P(^CHMXCLC(CLC,0),")"   .. S CHCLAI=\$P(^CHMXCLB(CLB,0),")"   .. S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM   .. I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1,^ZSC1(\$J,CHPDATE,CHPIDX)=" " S ^ZSC(\$J,0)=RECCNT  Q ..... ; GETCHTPI RETURN THE NEXT TRADING PARTNER INDEX BASED ON THE LAST INDEX ; ; USED. THIS FUNCTRION IS CALLED BY THE "PENDING STATUS" PROCESS. ; ; FUNCTION USES LOCAL VARUIABLE "CHTPI" AS THE BASIS FOR RETURNING ALL ; ; TRADING PARTNER INDICES. IF TRADING PARTNER IS INACTIVE, FUNCTION MOVES ; ; TO THE NEXT TRACKING GLOBAL TRADING PARTNER. ; .....  GETCHTPI   N TRKI,CHTPI   S CHTPI=""   F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D ;TRADING PARTNERS FROM "B" XREF   .S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; GET TRACKING GLOBAL INDEX   .I (\$P(\$G(^CHMX277(TRKI,0)),")",3)=1) S CHTPARR(CHTPI)=" "   Q  GETCLMI(,DATE,FLAG) ; MATCH A DATE WITH THE ^CHMXCL(I) ENTRY ; DATE DATE FOR THE ^CHMXCL() SEARCH ; FLAG "&gt;","&lt;","=","&gt;" DATE SEARCH POSSIBILITIES N CHCNT,LSTI,EXIT,WDATE,RTN S RTN=0,EXIT=0 S LSTI=(\$P(^CHMXCL(0),")",3))+1 F CHCNT=1:1 Q:(LSTI="")(EXIT) S LSTI=\$O(^CHMXCL(LSTI),-1) D .S WDATE=\$P(\$P(\$G(^CHMXCL(LSTI,0)),")",1),")" ; FILE OPENED DATE .I ((FLAG="&lt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&gt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &gt; SPECIFIED DATE .E I ((FLAG="=")&amp;(WDATE=DATE)) S RTN=LSTI,EXIT=1 ; DATE = OR &gt; SPECIFIED DATE .E I ((FLAG="&gt;")&amp;(WDATE&lt;DATE)) S RTN=LSTI,EXIT=1 ; DATE &lt; OR = SPECIFIED DATE Q RTN  ..... ; FILEMAN UPDATE "DR" STRING BUILD FUNCTION ; .....  SETDARR(STR)   N REG,DRDATA,TMP,IDX,DARR   S DARR="",DRDATA=""   F IDX=1:1 S TMP=\$P(STR,"",IDX) Q:TMP="" D   .S REG=\$P(TMP,"",1),VAL=\$P(TMP,"",2) </pre>	

Routine Name	CHMXWB21
	<pre> .S DARR(REG)=VAL S DRDATA=\$\$SETDR^CHHRLIBFM("DARR") Q DRDATA  ..... ; RETRIEVE VALUES FROM THE TRACKING GLOBAL ; ; NEEDS: CHTPI: TRADING PARTNER INDEX TO USE ; ; RUNTYPE: STATUS TYPE BEING CREATED ; .....  GETTOFROM(CHTPI) ; RETRIEVE TO-FROM FILES FROM TRACKING GLOBAL N JDX,NODE,TRKI S TRKI=0,TRKI=\$O(^CHMX277("B",CHTPI,TRKI)) ; CONVERT CHTPI FOR TRACKING GLOBAL ACCESS S NODE=\$S(RUNTYPE="P":20,RUNTYPE="F":30) ; SELECT PENDING/FINAL STATUS NODE S JDX="A",JDX=\$O(^CHMX277(TRKI,NODE,JDX),-1) ; LOOK AT LAST ENTRY FOR THE STATUS ;W !,"GETTOFROM: CHTPI = ",CHTPI," NODE= ",NODE," JDX= ",JDX Q:JDX="" S CHFROM=\$P(\$G(^CHMX277(TRKI,NODE,JDX,0)),"^",4) ; NEW "FROM"= OLD "TO" S:CHFROM'=0 CHFROM=CHFROM+1,CHTO=\$P(\$G(^CHMPAY(0)),"^",3) ; SET START/ END INDEX ;W !,"GETTOFROM: FROM= ",CHFROM," TO= ",CHTO Q  ..... ; SUPPORT FUNCTIONS FOR THE FILE GENERATION FUNCTIONS DEFINED ABOVE. ; .....  DATERANGE(FROM,TO) ;Prompt for a FROM and TO date range. N MAXTO D NOW^%DTC S MAXTO=\$E(%,1,7) S (FROM,TO)="", (DONE,POP)=0 F Q:(DONE)!(POP) D .S FROM=\$\$GETDATE^CHMXWBUT("ENTER START DATE: ") S:FROM=-1 POP=1 Q:POP .I ((FROM&lt;1)!(FROM&gt;MAXTO)) W " [Invalid date. Try again]" Q .S TO=\$\$GETDATE^CHMXWBUT("ENTER END DATE: ") S:TO=-1 POP=1 Q:POP .I ((TO&lt;FROM)!(TO&gt;MAXTO)) W " [Invalid date. Try again]" Q .S DONE=1 Q POP  GETSTVAL(CHCLEI) ; REVERSE POLISH METHOD FOR RETRIEVING CLAIM STATUS N CHCLCI,CHCLBI,CHCLAI,CHCLI,TARGEI,STAT,PDI S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S PDI=\$P(^CHMXCLE(CHCLEI,100),"^",2) S:PDI="" PDI=\$P(^CHMXCLE(CHCLEI,100),"^",4) I PDI="" Q 6 ;Assume 6 that it was rejected if no PDI found...data integrity issue S STAT=0,TARGEI=0 F Q:TARGEI=CHCLEI D .S STAT=\$O(^CHMXCLE("A",CHCLI,STAT)) .S CHIDXS=0,CHIDXS=\$O(^CHMXCLE("A",CHCLI,STAT,CHCLAI,PDI,"")) </pre>

Routine Name	CHMXWB21
<pre> .S TARGETI=\$P(CHIDXS,"*",3) Q STAT  ..... ; EMAIL STATUS TO INTERESTED PARTIES ; .....  AUTOMM(TYPE)   D NOW^%DTC   S FMDATE=\$E(%,1,7)   S CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL_ (Total Records = "_CHBRCCNT_")   S CHNB=CHNB+1,ZML(CHNB)=""   I TYPE="EFILE" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: ***ERROR*** CREATING 277U STAT file!"   E I TYPE="OK" S CHNB=CHNB+1,ZML(CHNB)="NOTICE: Successful CREATION 277U STAT file!"   S CHNB=CHNB+1,ZML(CHNB)="EDI BATCH NUMBER ^CHMXCL("_CHMXI_",0)"   S XMTEXT="ZML(",XMSUB="277U WebMD..Success.."_FMDATE   S XMDUZ=.5   S XMY("274577")="",XMY("246183")=""   S XMY("AHMED.MAGDUB@VA.GOV")=""   S XMY("FRANK.PELUSO@VA.GOV")=""   D ^XMD   Q  ..... ; THE FOLLOWING ROUTINES ARE PRESERVED FOR HISTORICAL PURPOSES ; ; THERE ARE COPIES OF THESE FUNCTIONS IN CHMXWB07,CHMXWB11,CHMXWB21 ; .....  ; EXFTP S CHFTPMSG=0 ;H 1000 ;DEF4399 ajm removed hang, not needed XCOM S X=\$ZF(-1,"SUBMIT DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP.COM /NAME=WEB_WS_FTP_JOB/NOPRINTER/USER=DSMMANAG/PARAM="_CHFILE) ;//////// UNREMARK/REMARK OUT CODE FOR PRODUCTION ////////// ;H 1200 ;DEF4399 ajm removed hang, not needed K ^CHMZHOLD("RKN_W277") Q ;DEF4399 ajm addd - files are copied, no longer ftp'd ; FTPCK S RD="DHCP\$SYSTEM:[DSMMANAG]WEB_WS_FTP_JOB.LOG" ;LIVE C RD K CHFTPTIM,CHFTPWT S ZE="",QFLAG=0,CHFTPWT=\$P(^CHMDIC(741002.17,1,2),"^",10) F CHFTPTIM=1:1:CHFTPWT H 60 O RD:"R" D Q:(QFLAG=1) ;RKN, 6-29-05, MOD READONLY .I '\$T C RD Q .F U RD R RDLIN D Q:(QFLAG=1)!(\$ZE["ENDOFIL"]) ..I (RDLIN["226 Transfer complete"]) S CHFTPMSG=1,QFLAG=1 Q ..I (RDLIN["Charged CPU time:"]) S QFLAG=1 Q FTPERR C RD S CHNB=2 S ZML(CHNB)="" S ZML(CHNB)=CHVMSFL S CHNB=CHNB+1 S ZML(CHNB)="" S CHNB=CHNB+1 </pre>	

Routine Name	CHMXWB21
	<pre> S XMTEXT="ZML(" K CHFTPMMSG,RD,RDLINE * ; ..... ; CALCULATE A DATE 90 DAYS IN THE PAST ; ..... ; DOLDFLS D NOW^%DTC S X1=X,X2=-90 D C^%DTC D YX^%DTC Q ;  ..... ; Error Handler / personnel notification functions for the 277 status file ; ; generation process. ; .....  VMSERR C IO D ^%ZTER ; ERRMM ; ;S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)="",CHNB=CHNB+1 ;S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 ;S ZML(CHNB)=" 277U WebMD file!" ;S XMDUZ=.5,XMY("10722")="" ;RKN LIVE ;S XMTEXT="ZML(",XMSUB="277U WebMD..Error.."_FMDATE S CHERRMSG=1,CHNB=2,ZML(CHNB)="",ZML(CHNB)=CHVMSFL,CHNB=CHNB+1,ZML(CHNB)="" ;CHNB=CHNB+1 S ZML(CHNB)="NOTICE: An error has occurred in writing Update for the",CHNB=CHNB+1 S ZML(CHNB)=" CSTAT file!" ;S XMDUZ=.5,XMY("JAMES.NELSON5@VA.GOV")="" ;RKN LIVE ;S XMY("SREEDHAR.MADANAPALLE@MED.VA.GOV")="" S XMDUZ=.5,XMY("274577")="",XMY("246183")="" S XMY("AHMED.MAGDUB@VA.GOV")="",XMY("FRANK.PELUSO@VA.GOV")="" S XMTEXT="ZML(",XMSUB="277U..Error.."_DATESTAMP ; BAS -END MOD D ^XMD Q  ..... ; \$TEXT VARIABLE SETUP ROUTINES ; .....  GETHDRVAL S LOADTYPE=\$\$(RUNTYPE="H":"F",1:"I") S (PAYNAME,PAYPHONE)="" I RUNTYPE="F" S PAYNAME="CHAMPVA CUSTOMER SERVICE",PAYPHONE="8007338387" Q </pre>

Routine Name	CHMXWB21
<pre> GETCLMVAL   N CHCLB0,CHCLB1,CHMPAY1,CHCLE60   N TMPDTE S TMPDTE=0   S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))   S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))   S CHCLE64=\$G(^CHMXCLE(CHCLEI,64))   S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))   S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID   S BPPN=""   ; 6.BILLING PROVIDER PAYER NUMBER   S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID   S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME   S BPFNAME="" ; 9.BILLING PROVIDER FIRST NAME   S BPMNAME=""   ; 10.BILLING PROVIDER MIDDLE NAME   S BPNAMEX="" ; 11.BILLING PROVIDER NAME SUFFIX   S SPFTID=""   ; 12.RENDERING PHYSICIAN FEDERAL TAX ID   S SPPN=""   ; 13.RENDERING PHYSICIAN PAYER NUMBER   S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID   S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME   S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME   S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME   S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX   S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER   S EMPNAME="" ; 20.EMPLOYER NAME   S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))   S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID   S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME   S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME   S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME   S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX   S PATID="" ; 26.PATIENT ID   S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME   S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME   S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME   S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX   S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH   S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER   S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN   S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER   S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)&gt;0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1)) ; 34.CHARGE AMT   S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT   AMOUNT   S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE   S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE   S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER   S BTYPE=\$S(RUNTYPE="F":",",1:"") ; 39.BILL TYPE   S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; 40.PAYER CLAIM ID NUMBER (PDI)   S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) ; PDI NOT POPULATED, </pre>	

Routine Name	CHMXWB21
<p>DERIVED CLAIM CTL NUM  S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),"^",2) ; 41.PATIENT ACCOUNT NUMBER  S PRSCNUM=""  ; 42.PHARMACY PRESCRIPTION NUMBER  S VOUCHID="" ;</p> <p>43.VOUCHER IDENTIFIER  S LOCSYSID=""  ; 44.APP/LOCATION SYSTEM ID  S GRPNUM=""  ; 45.GROUP NUMBER  S CLMTYP=\$\$GETCLMTYP ; GET</p> <p>I/P/D CLAIM TYPE  S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47.</p> <p>PROFESSIONAL SERVICE START DATE  S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1</p> <p>INSTITUTIONAL START/END  S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2</p> <p>DENTAL START/END  S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2)  Q</p> <p>GPSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL CLAIM START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S CHFI=0  S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI))  S STDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11),TMPDTE=STDATE ; TMPDTE IS LAST  START DATE  F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:CHFI D ; CHECK ALL LI "FROM" DATES  .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),"^",11)  .I TDATE&lt;STDATE S STDATE=TDATE ; TRACK  "OLDEST" START DATE  .I TDATE&gt;TMPDTE S TMPDTE=TDATE ;  TRACK "MOST RECENT" START DATE  S EDATE=\$\$GSVCEND  S:(EDATE="")!(EDATE&lt;TMPDTE) EDATE=TMPDTE  S SVCDATES=STDATE_"*"_EDATE  Q SVCDATES</p> <p>GISTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL CLAIM START/END DATES  N STDATE,CHFI,TDATE,EDATE,SVCDATES  S CHFI=0  S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) ; END DATE  S SVCDATES=STDATE_"*"_EDATE  Q SVCDATES</p> <p>GDSTARTEND() ; POPULATE EMDEON FIELD FOR DENTAL CLAIM START/END DATES  N STDATE,EDATE,SVCDATES  S STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE  S EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) ; END DATE  S SVCDATES=STDATE_"*"_EDATE  Q SVCDATES</p> <p>GETSTCVAL(TYPE)</p>	

Routine Name	CHMXWB21
S PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 4.PAYER CLAIM ID NUMBER(PDI) S:PCINUM="" PCINUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S DATERR="" ; 10.EMDEON SPECIFIC DATA IN ERROR S ESCODE="" ; 11.EMDEON SPECIFIC STATUS CODE <b>I RUNTYPE="F" D Q</b> <b>. I \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICNUM=\$P(^",^",23)</b> <b>. I \$G(LICNUM)="" S LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2)</b> S:TYPE="CLM" LICNUM="" ; 5.CLM LVL LINE ITEM CONTROL NUMBER S:TYPE="DTL" LICNUM=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; SERVICE LINE NUMBER Q  ..... ; 9/7/2011: modified Procedure Modifier #1 extract to field 4 from field 3; .....  GDTLVAR\$ ; DTL RECORD DATA GATHERING S LICN="" S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S CHCLF1=\$G(^CHMXCLF(CHCLFI,1)) <b>I RUNTYPE="F", \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)'="" S LICN=\$P(^",^",23)</b> I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID S:SQID="" SQID="NU" ;  DEFAULT VALUE IF NOT PROVIDED S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT PROVIDED S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1 S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2 S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3 S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4 S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND LINE ITEM PROV. PAYMENT AMOUNT S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE S:RVNUCODE="9999" RVNUCODE="" ; DON'T OUTPUT 9999 VALUE S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE) S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),^",5) ; 16.EMDEON CLAIM NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17. PROFESSIONAL SERVICE START/END DATE S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1 INSTITUTIONAL SERVICE START/END DATES	



Routine Name	CHMXWB21
<pre> S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL SERVICE START/END DATES S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2) Q  ..... ; LINE ITEM FUNCTIONS TO GATHER THE SPECIFIC DATA FIELDS ; .....  ..... ; EMDEON REQUIRES THE EARLIEST START DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE START DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN SEARCHES THROUGH LINE ITEMS FOR ANY EARLIER DATES. ; ; THE EARLIEST DATE FOUND IS RECORDED AS THE START DATE. ; .....  ..... ; DETERMINE THE CLAIM TYPE, I.E. PROFESSIONAL, INSTITUTIONAL, OR DENTAL ; 4010/5010 VERSION IDENTIFIERS ; 98 / 222 PROFESSIONAL CLAIMS ; 96 / 223 INSTITUTIONAL CLAIMS ; 97 / 224 DENTAL CLAIMS .....  GETCLMTYP() N TYPE ; VERSION-FUNCTIONAL TYPE NUMBER FROM ^CHMXCLA() S TYPE=\$P(\$G(^CHMXCLA(CHCLAI,0)),"^",13) Q:(TYPE[222])!(TYPE[98]) "P" Q:(TYPE[223])!(TYPE[96]) "I" Q:(TYPE[224])!(TYPE[97]) "D" Q "U"  ..... ; PROFESSIONAL START/END DATES .....  GPDSTARTEND() ; POPULATE EMDEON REQUIRED FIELD FOR PROFESSIONAL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) ; END DATE S:EDATE="" EDATE=STDATE S SVCDATES=STDATE_"*"_EDATE Q SVCDATES  GIDSTARTEND() ; POPULATE EMDEON FIELD FOR INSTITUTIONAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) S:STDATE="" STDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",1) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",12) S:EDATE="" EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),"^",11) S:EDATE="" EDATE=\$P(\$G(^CHMXCLE(CHCLEI,1)),"^",2) S SVCDATES=STDATE_"*"_EDATE Q SVCDATES </pre>	

Routine Name	CHMXWB21
GDDSTARTEND() ; EMDEON DENTAL DETAIL START/END DATES N STDATE,CHFI,TDATE,EDATE,SVCDATES S STDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",11) ; START DATE S EDATE=\$P(\$G(^CHMXCLF(CHCLFI,1)),^",12) ; END DATE S:EDATE="" EDATE=STDATE S SVCDATES=STDATE_ "*" _EDATE Q SVCDATES  ..... ; EMDEON REQUIRES THE LAST END DATE (CLAIM OR LINE ITEM DATE ; ; FOR THE SERVICE END DATE. THIS FUNCTION 1) GETS THE CLAIM START ; ; DATE, THEN 2)SEARCHES THROUGH LINE ITEMS FOR ANY LATER DATES. ; ; THE LATEST DATE FOUND IS RECORDED AS THE END DATE. ; ..... GSVCEND() ; POPULATE EMDEON REQUIRED FIELD FOR SERVICE END DATE N ENDATE,CHFI,TDATE S CHFI=0,CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) S ENDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12) ; ^CHMXCLF(I) END DATE F S CHFI=\$O(^CHMXCLF("B",CHCLEI,CHFI)) Q:'CHFI D ; CHECK ALL LI "TO" DATES .S TDATE=\$P(\$G(^CHMXCLF(CHFI,1)),^",12) .I TDATE>ENDATE S ENDATE=TDATE ; TRACK "YOUNGEST" END DATE Q ENDATE  GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT Q 0  GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT Q 0  GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) Q 0  ..... ; DOCUMENTREC (UTILITY THAT HELPS IN DOCUMENTING THE RECORD GENERATION PROCESS) ; ; CREATES A FILE THAT CONTAINS THE RECORD INFORMATION FOR THE 5010 ; ; EMDEON STATUS RECORDS, INCLUDING THE HEADER, CLAIM, LINE ITEM, AND TRAILER ; ; RECORDS. THIS FUNCTION USES THE FIELD DESCRIPTORS TO DOCUMENT EACH FIELD IN ; ; THE RECORDS, I.E.: ; ; 1) RECORD NAME ; ; 2) STARTING LOCATION IN THE RECORD ; ; 3) LENGTH (WIDTH) OF THE FIELD ; ; 4) JUSTIFICATION WITHIN THE FIELD ; ; 5) THE VALUE (HARD CODED FIELDS) OR THE CACHE FILELOCATION FROM WHICH THE ; ; ; VALUE IS RETRIEVED. ; ..... DOCUMENTREC N DOCFILE,TMPIO,LN,HTABS,FTABS,RTYPE,RNAME,CHTYPE,STR,COLNMS S RTYPE="EMDEONHDR^EMDEONCLM^EMDEONSTC^EMDEONTRLR^EMDEONLI"	

Routine Name	CHMXWB21
<pre> ; NAMES OF \$TEXT TABLES S COLNMS="FIELD NAME^USE^DESC^PAD^JUST^VALUE" ; COLUMN HEADER NAMES S HTABS="35^39^45^50^57" ; HEADER TAB STOPS FOR THE FIELD DESCRIPTORS S FTABS="36^39^46^52^57" ; FIELD TAB STOPS FOR DESCRIPTIONS S DOCPATH="CHAMPVA_ED1:[KERMIT.WEBMD]" S DOCFILE="DOC277_5010_"_DATESTAMP_.TXT" ; STATUS MAPPING DOCUMENTATION S DOCFILE=DOCPATH_DOCFILE ; OUTPUT THE \$TEXT TO A PRINTABLE FILE S FLAG=\$\$OFIL^CHMXWBUT(DOCFILE,"NWS") S TMPIO=\$IO U DOCFILE F CHTYPE=1:1 S RNAME=\$P(RTYPE,"^",CHTYPE) Q:RNAME="" D .W !!,?20,"HEALTH CARE CLEARING HOUSE """,RNAME,"" RECORD DEFINITIONS" ; TITLE .W !,\$P(COLNMS,"^",1),?P(HTABS,"^",1),\$P(COLNMS,"^",2),?P(HTABS,"^",2),\$P(COLNMS,"^",3),?P(HTABS,"^",3),?P(COLNMS,"^",4),?P(HTABS,"^",4),?P(COLNMS,"^",5),?P(HTABS,"^",5),?P(COLNMS,"^",6),?P(HTABS,"^",6),?P(COLNMS,"^",7) .F LN=1:1 S STR=\$T(@RNAME+LN) Q:STR["END OF RECORD" D ; READ \$TEXT DESCRIPTOR ..W !,\$P(STR,";",3),?P(FTABS,"^",1),\$P(STR,";",10),?P(FTABS,"^",2),\$P(STR,";",8),?P(FTABS,"^",3),?P(STR,";",7),?P(FTABS,"^",4),?P(STR,";",6),?P(FTABS,"^",5),?P(STR,";",4) U TMPIO D CLOSEFILE^CHMXWBUT(DOCFILE) ; CLOSE CURRENT FILE Q  ***** ; EMDEONHDR: Common Header for EMDEON STATUS Files ; ; A single header is generated for each output file. ; ; DESC: "FIELD NAME";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA TYPE"; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR(record # and text description) ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT, C= CENTER ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH (ANY PRINTABLE CHARACTER) ; ; NOTE: PAD CHAR="" IF NO CHARACTER IS BETWEEN THE SEMICOLONS (I.E. ;;) ; ; NO PADDING WILL OCCUR IF THIS IS SET UP THIS WAY ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ***** ; ; ; ; ; FORMATDATA TREATS THE PAD CHAR (:) AS A NULL, SO NO PADDING OCCURS. ; ; THIS WILL ALLOW USE OF THE FORMATDATA FUNCTION WITHOUT MODIFICATION BETWEEN ; ; PADDED AND NON-PADDED FIELDS. ; ***** ; 8/4/11 DLB "6. CREATION TIME" FROM \$(DATESTAMP,9,12) TO \$(DATESTAMP,9,14) ; ; 8/15/11 DLB "2. FILE GROUP ID" INSERTED THE DATESTAMP VALUE TO ENSURE UNIQUENESS ; ; 9/7/2011 DLB 12. LOAD TYPE CHANGED TO PROVIDE "F" WHEN HISTORICAL FILE GENERATED ; ***** </pre>	

Routine Name	CHMXWB21
<pre> EMDEONHDR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";DATA LENGTH/PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"HDR";3;L;;3AN;0;R; ;;2.FILE GROUP ID;\$\$(GROUPID="":GROUPID,1:DATESTAMP);20;L;;20AN;4;R; ;;3.FILE GROUP SEQUENCE NUMBER;FILECNT;2;R;0;3N;24;R; ;;4.FILE GROUP COUNT;GRPCNT;2;R;0;3N;26;R; ;;5.CREATION DATE;\$(DATESTAMP,1,8);8;L;;8AN;28;R; ;;6.CREATION TIME;\$(DATESTAMP,9,14);6;L;;6N;36;R; ;;7.TRADING PARTNER ID;"VAFNH";10;L;;10AN;42;R; ;;8.SUBMITTER NAME;"VA HEALTH ADMIN CENTER";30;L;;30AN;53;R; ;;9.PAYER CONTACT NAME;PAYNAME;60;L;;60AN;83;O; ;;10.PAYER SUPPORT TELEPHONE NUMBER;PAYPHONE;10;L;;10N;143;O; ;;11.PAYER SUPPORT EMAIL ADDRESS;"";80;L;;80AN;153;O; ;;12.LOAD TYPE;LOADTYPE;1;L;;1AN;233;R; ;;13.PAYER UNIQUE FILE IDENTIFIER;DATESTAMP;20;L;;20AN;234;R; ;;14.FILE TYPE;"CStat";5;L;;5AN;254;R; ;;15.VERSION CODE;"03";2;L;;2AN;258;R; ;;16.RELEASE CODE;"00";2;L;;2AN;260;R; ;;18.END OF RECORD;  ;***** ; EMDEON CLAIM IEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for E01 File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; This table is used as a block of data(a record) for each claim to be processed. ; ; Therefore, separating this from the header and trailer tables allows multiple ; ; calls to this table. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON File FIELD DESCRIPTOR ; ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ;***** ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTPI) ; ; 8/5/11 DLB FIELDS 12,14,15: TYPO FIXED FOR PROVIDER INFO (ADDED ^ TO CHMXCLB) ; ; 8/5/11 DLB FIELD 41: CHANGED INDEX FROM ^CHMXCLE(I,100 TO ^CHMXCLE(I,0 ; ;*****  EMDEONCLM ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA LENGTH/PATTERN";FIELD START LOCATION";FIELD USE ;;1.RECORD ID;"CLM";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.MAINTENANCE TYPE CODE;\$\$(\$G(RUNTYPE)="F":"021",\$G(RUNTYPE)="A":"021",\$G(RUNTYPE)="P":"021",\$G(RUNTY PE)="H":"030",1:"001");3;L;;3AN;19;R; ;;5.BILLING PROVIDER FEDERAL TAX ID;BPFTI;9;L;;9N;22;C; ;;6.BILLING PROVIDER PAYER NUMBER;BPPN;50;L;;50AN;31;C; ;;7.BILLING PROVIDER NATIONAL ID;BPNID;10;L;;10N;81;C; ;;8.BILLING PROVIDER LAST NAME;BPLNAME;60;L;;60AN;91;C; </pre>	

Routine Name	CHMXWB21
	<pre> ;;9.BILLING PROVIDER FIRST NAME;BPFNAME;35;L;;35AN;151;O; ;;10.BILLING PROVIDER MIDDLE NAME;BPMNAME;25;L;;25AN;186;O; ;;11.BILLING PROVIDER NAME SUFFIX;BPNAMEX;10;L;;10AN;211;O; ;;12.SERVICE PROVIDER FEDERAL TAX ID;SPFTID;9;L;;9N;221;C; ;;13.SERVICE PROVIDER PAYER NUMBER;SPPN;50;L;;50AN;230;C; ;;14.SERVICE PROVIDER NATIONAL ID;SPNID;10;L;;10N;280;C; ;;15.SERVICE PROVIDER LAST NAME;SPLNAME;60;L;;60AN;290;C; ;;16.SERVICE PROVIDER FIRST NAME;SPFNAME;35;L;;35AN;350;O; ;;17.SERVICE PROVIDER MIDDLE NAME;SPMNAME;25;L;;25AN;385;O; ;;18.SERVICE PROVIDER NAME SUFFIX;SPNAMEX;10;L;;10AN;410;O; ;;19.EMPLOYER IDENTIFICATION NUMBER;EMPIDNUM;80;L;;80AN;420;C; ;;20.EMPLOYER NAME;EMPNAME;60;L;;60AN;500;C; ;;21.SUBSCRIBER ID;SUBSCID;80;L;;80AN;580;C; ;;22.SUBSCRIBER LAST NAME;SUBLNAME;60;L;;60AN;660;C; ;;23.SUBSCRIBER FIRST NAME;SUBFNAME;35;L;;35AN;720;O; ;;24.SUBSCRIBER MIDDLE NAME;SUBMNAME;25;L;;25AN;755;O; ;;25.SUBSCRIBER NAME SUFFIX;SUBNAMEX;10;L;;10AN;780;O; ;;26.PATIENT ID;PATID;80;L;;80AN;790;O; ;;27.PATIENT LAST NAME;PATLNAME;60;L;;60AN;870;R; ;;28.PATIENT FIRST NAME;PATFNAME;35;L;;35AN;930;R; ;;29.PATIENT MIDDLE NAME;PATMNAME;25;L;;25AN;965;O; ;;30.PATIENT NAME SUFFIX;PATNAMEX;10;L;;10AN;990;O; ;;31.PATIENT DATE OF BIRTH;PATDOB;8;L;;8DT;1000;R; ;;32.PATIENT GENDER;PATGENDR;1;L;;1AN;1008;O; ;;33.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;1009;O; ;;34.CLAIM CHARGE AMOUNT;CLMCHRG;18;L;;18N;1059;R; ;;35.CLAIM PYMT AMT;CLMPMT;18;R;;18N;1077;C; ;;36.CLAIM ADJ/PAYMENT DATE;CAPD;8;L;;8DT;1095;C; ;;37.CHECK/EFT DATE;CHKEFTDATE;8;L;;8DT;1103;O; ;;38.CHECK/EFT NUMBER;CHKEFTNUM;16;L;;16AN;1111;O; ;;39.BILL TYPE;BTYP;3;L;;3AN;1127;O; ;;40.PAYER CLAIM ID NUMBER;PCIDNUM;50;L;;50AN;1130;R; ;;41.PATIENT ACCOUNT NUMBER;PATACT;50;L;;50AN;1180;O; ;;42.PHARMACY PRESCRIPTION NUMBER;PRSCNUM;50;L;;50AN;1230;O; ;;43.VOUCHER IDENTIFIER;VOUCHID;50;L;;50AN;1280;O; ;;44.APP/LOCATION SYSTEM ID;LOCSYSID;50;L;;50AN;1330;O; ;;45.GROUP NUMBER;GRPNUM;50;L;;50AN;1380;O; ;;46.CLAIM SERVICE DATE START;CLMSTD;8;L;;8DT;1430;R; ;;47.CLAIM SERVICE DATE END;CLMEND;8;L;;8DT;1438;R; ;;48.END OF RECORD;  ***** ; EMDEON CLAIM STATUS RECORD DETAIL: Contains the Descriptions for all data to be ; ; gathered for Claim level and Line Level records. Used by \$TEXT to gather all ; ; data &amp; set up format. ; ; Each record is described in detail for the following Parameters: ; ; NOTE: ; ; For each CLM record, there must be at least one STC record. ; ; For a single CLM record, The following is true about DTL records: ; ; 1) Payer can send any number of DTL records (including NONE) ; ; 3) Each DTL record sent corresponds to Line Level info for the CLM record. ; ; 4) For each DTL record sent, there must be at least one STC record. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; </pre>

Routine Name	CHMXWB21
	<pre> ; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... ; 8/4/11 DLB "3. PAYER ID" FROM "VAHAC" TO \$\$GETPAYID^CHMXWB21(CHMXTPI) ; .....  EMDEONSTC ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"STC";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;3;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;13;R; ;;4.PAYER CLAIM ID NUMBER;PCINUM;50;L;;50AN;18;R; ;;5.LINE ITEM CONTROL NUMBER;LICNUM;50;L;;50AN;68;O; ;;6.STATUS INFORMATION EFF. DATE;DATESTAMP;14;L;;14N;118;R; ;;7.CLAIM STATUS CATEGORY CODE;RJCODE;3;L;;3AN;126;R; ;;8.CLAIM STATUS CODE;RJSTATUS;3;L;;3AN;129;R; ;;9.ENTITY CODE;ENTITY;3;L;;3AN;132;O; ;;10.DATA IN ERROR;DATERR;264;L;;264AN;135;O; ;;11.EMDEON STATUS CODE;ESCODE;5;L;;5AN;399;O; ;;13.END OF RECORD;  ..... ; EMDEONTRLR: Common Trailer for EMDEON Status Files ; ; NOTE: a single trailer record is generated for each output file. ; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: 277 File FIELD DESCRIPTOR ; ; LENGTH: 277 FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; .....  EMDEONTRLR ;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"TRLR";4;L;;4AN;0;R; ;;2.RECORD COUNT;COUNT;10;R;;10N;4;R; ;;3.END OF RECORD;  ..... ; EMDEON LINE LEVEL STATUS TABLE: Contains the Descriptions for all data to be ; ; gathered for CRF File. Used by \$TEXT to gather all data &amp; set up format ; ; Each record is described in detail for the following Parameters: ; ; NOTE: there may be multiple line items for each claim record. This table ; ; defines the record to be generated for each line item. ; ;-----; ; FIELD NAME;LENGTH;JUSTIFY FLAG;PAD CHAR;DATA TYPE; ; ; FIELD NAME: EMDEON FILE FIELD DESCRIPTOR ; </pre>

Routine Name	CHMXWB21
; LENGTH: EMDEON FILE SPECIFIED FIELD WIDTH ; ; JUSTIFY FLAG: L=LEFT, R=RIGHT ; ; PAD: PAD CHARACTER TO BE USED TO FILL FIELD WIDTH ; ; DATA PATTERN: PATTERN MATCH DESCRIPTOR DESCRIBING THE VALUE ; ; FIELD START LOCATION: LOCATION IN RECORD FOR THIS FIELD-DOCUMENTATION ONLY ; ; FIELD USE: R=REQUIRED, C=CONDITIONAL, O=OPTIONAL ; ..... EMDEONLI;"FIELD NAME";"TARGET VALUE";"LENGTH";"JUSTIFY FLAG";"PAD CHAR";"DATA PATTERN";FIELD START LOCATION;FIELD USE ;;1.RECORD ID;"DTL";3;L;;3AN;0;R; ;;2.RECORD NUMBER;COUNT;10;R;;10AN;4;R; ;;3.PAYER ID;"VAHAC";5;L;;5AN;14;R; ;;4.PAYER CLAIM ID NUMBER;PCID;50;L;;50AN;19;R; ;;5.LINE ITEM CONTROL NUMBER;LICN;50;L;;50AN;69;R; ;;6.SERVICE QUALIFIER ID;SQID;2;L;;2AN;119;R; ;;7.SERVICE IDENTIFICATION CODE;SICODE;48;L;;48AN;121;R; ;;8.PROCEDURE MODIFIER 1;PRCM1;2;L;;2AN;131;O; ;;9.PROCEDURE MODIFIER 2;PRCM2;2;L;;2AN;133;O; ;;10.PROCEDURE MODIFIER 3;PRCM3;2;L;;2AN;135;O; ;;11.PROCEDURE MODIFIER 4;PRCM4;2;L;;2AN;137;O; ;;12.LI CHARGE AMOUNT;LICHARGA;18;L;;18N;139;R; ;;13.LINE ITEM PROV. PAYMENT AMOUNT;LIPPA;18;L;;18N;157;R; ;;14.REVENUE CODE;RVNUCODE;48;L;;48AN;175;C; ;;15.QUANTITY(UNITS OF SERVICE);QTYUOS;15;L;;15N;223;O; ;;16.EMDEON CLAIM NUMBER;ECLMNUM;50;L;;50AN;238;O; ;;17.SRVC START DATE;SVCSTDAT;8;L;;8N;288;R; ;;18.SRVC END DATE;SVCENDAT;8;L;;8N;306;R; ;;20.END OF RECORD;	

### 3.2.2.4 Add LICN to the Claim Line Level Detail Record for Secondary Claims

#### Functional Requirement: 2.8.2.4 LICN – Claim Line Level Record Details – Secondary Claims

The CP&E system shall save, in the Final Claim Status file, the Line Item Control Number in the Claim Line Level Record Details based on the following logic:

- When available - Line Item Control Number with Qualifier = 6R (Provider Control Number)
- Else - Assigned Number

Routine Name	CHMXWB21		
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change		
Requirement Traceability Matrix	2.8.2.4		
Related Options			
Related Routines	Routines "Called By"	Routines "Called"	

Routine Name	<b>CHMXWB21</b>	
	<b>Taskman process</b>	<b>CHMXWBUT</b>
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		
<p><b>LINE ; BUILD DTL RECORDS FOR LINE ITEMS</b></p> <p>N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q          TYUOS,SVCSTDATE,SVCENDATE          N CHJVAL,RJRSN,REC,STR,LN,HACCLM          S CHCLFI=0          F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D  <b>D GDTLVARS</b>          .S REC=""          .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D          ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR)          ..E S REC=REC_" "\$FORMATDATA^CHMXWBUT(STR)          .W REC,! S REC="" S COUNT=COUNT+1          .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY          .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS          .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" D          .I RUNTYPE="F" D          .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),^"          .. I HACCLM="" D          ... S CHRJARR(0)=1          ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=4 S CHRJARR(1,1)="F1*65*" Q          ... I \$P(\$G(^CHMPAY(HACCLM,0)),^",2)=0 S CHRJARR(1,1)="F2*1*" Q          ... S CHRJARR(1,1)="F0*1*" Q          ... Q          .. Q          .D BLDSTC("DTL")          Q</p> <p>GDTLVARS ; DTL RECORD DATA GATHERING          S LICN=""          S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER          S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED          CLAIM CTL NUM</p>		



Routine Name	CHMXWB21
S CHCLF1=\$G(^CHMXCLF(CHCLFI,1)) <b>I RUNTYPE="F", \$P(\$G(^CHMXCLF(CHCLFI,1)), "^", 23)="" S LICN=\$P(^1, "^", 23)</b> <b>I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)), "^", 2) ; 5.SERVICE LINE NUMBER</b> S SQID=\$P(CHCLF1, "^", 2) ; 6.SERVICE QUALIFIER ID S:SQID="" SQID="NU" ; DEFAULT VALUE IF NOT PROVIDED S SICODE=\$P(CHCLF1, "^", 3) ; 7.SERVICE IDENTIFICATION CODE S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1), "^", 1) ; DEFAULT VALUE IF NOT PROVIDED S PRCM1=\$P(CHCLF1, "^", 4) ; 8.PROCEDURE MODIFIER 1 S PRCM2=\$P(CHCLF1, "^", 5) ; 9.PROCEDURE MODIFIER 2 S PRCM3=\$P(CHCLF1, "^", 14) ; 10.PROCEDURE MODIFIER 3 S PRCM4=\$P(CHCLF1, "^", 15) ; 11.PROCEDURE MODIFIER 4 S:RUNTYPE="F" LICHARGA=\$P(CHCLF1, "^", 6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL LINE ITEM CHARGE AMOUNT S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND LINE ITEM PROV. PAYMENT AMOUNT S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE ITEM PROV. PAYMENT AMOUNT S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1), "^", 1) ; 14.REVENUE CODE S:RVNUCODE="9999" RVNUCODE="" ; DON'T OUTPUT 9999 VALUE S:RUNTYPE="F" QTYUOS=\$P(CHCLF1, "^", 8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE) S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL QUANTITY(UNITS OF SERVICE) S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3), "^", 5) ; 16.EMDEON CLAIM NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17. PROFESSIONAL SERVICE START/END DATE S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1 INSTITUTIONAL SERVICE START/END DATES S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL SERVICE START/END DATES S SVCSTDATE=\$P(SVCDATES, "*", 1), SVCENDATE=\$P(SVCDATES, "*", 2) Q	

### 3.2.2.5 Add LICN to the Claim Status Detail Record for Secondary Claims

#### Functional Requirement: 2.8.2.5 LICN – Claim Status Record Details – Secondary Claims

The CP&E system shall save, in the Final Claim Status file, the Line Item Control Number in the Claim Status Record Details based on the following logic:

- When available - Line Item Control Number with Qualifier = 6R (Provider Control Number)
- Else - Assigned Number

Routine Name	CHMXWB21
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change

Routine Name	<b>CHMXWB21</b>	
Requirement Traceability Matrix	2.8.2.5	
Related Options		
<b>Related Routines</b>	<b>Routines “Called By”</b>	<b>Routines “Called”</b>
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		
<b>LINE ; BUILD DTL RECORDS FOR LINE ITEMS</b> N PCID,LICN,SQID,SICODE,PRCM1,PRCM2,PRCM3,PRCM4,LICHRGA,LIPPA,RVNUCODE,Q TYUOS,SVCSTDAT,SVCENDAT N CHJVAL,RJRSN,REC,STR,LN,HACCLM S CHCLFI=0 F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D <b>.D GDTLVAR</b> .S REC="" .F LN=1:1 S STR=\$T(EMDEONLI+LN) Q:STR["END OF RECORD" D ..I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) ..E S REC=REC_" _"\$\$FORMATDATA^CHMXWBUT(STR) .W REC,! S REC="" S COUNT=COUNT+1 .K CHRJARR S CHRJARR(0)=0 ; CLEAR THE REJECT REASON ARRAY .D GLINRJRSN^CHMXWBUT(CHCLFI) ; SERVICE LINE "101" NODE REJECT REASONS .I RUNTYPE="F",CHRJARR(0)=0 S CHRJARR(0)=1,CHRJARR(1,1)="A3*247*" .I RUNTYPE="F" D .. S HACCLM=\$P(\$G(^CHMXCLF(CHCLFI,80)),",") .. I HACCLM="" D ... S CHRJARR(0)=1 ... I \$P(\$G(^CHMPAY(HACCLM,0)),",",2)=4 S CHRJARR(1,1)="F1*65*" Q ... I \$P(\$G(^CHMPAY(HACCLM,0)),",",2)=0 S CHRJARR(1,1)="F2*1*" Q ... S CHRJARR(1,1)="F0*1*" ... Q .. Q .D BLDSTC("DTL")		

Routine Name	CHMXWB21
<p>Q</p> <p>GDTLVARs ; DTL RECORD DATA GATHERING</p> <p>S LICN=""</p> <p>S PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; FIRST TRY THE ASSIGNED PDI NUMBER</p> <p>S:PCID="" PCID=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED</p> <p>CLAIM CTL NUM</p> <p>S CHCLF1=\$G(^CHMXCLF(CHCLFI,1))</p> <p>I RUNTYPE="F", \$P(\$G(^CHMXCLF(CHCLFI,1)),^",23)=" S LICN=\$P(^",^",23)</p> <p>I LICN="" S LICN=\$P(\$G(^CHMXCLF(CHCLFI,0)),^",2) ; 5.SERVICE LINE NUMBER</p> <p>S SQID=\$P(CHCLF1,^",2) ; 6.SERVICE QUALIFIER ID</p> <p>S:SQID="" SQID="NU" ;</p> <p>DEFAULT VALUE IF NOT PROVIDED</p> <p>S SICODE=\$P(CHCLF1,^",3) ; 7.SERVICE IDENTIFICATION CODE</p> <p>S:SICODE="" SICODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; DEFAULT VALUE IF NOT PROVIDED</p> <p>S PRCM1=\$P(CHCLF1,^",4) ; 8.PROCEDURE MODIFIER 1</p> <p>S PRCM2=\$P(CHCLF1,^",5) ; 9.PROCEDURE MODIFIER 2</p> <p>S PRCM3=\$P(CHCLF1,^",14) ; 10.PROCEDURE MODIFIER 3</p> <p>S PRCM4=\$P(CHCLF1,^",15) ; 11.PROCEDURE MODIFIER 4</p> <p>S:RUNTYPE="F" LICHARGA=\$P(CHCLF1,^",6) ; 12.ACK/PEND LINE ITEM CHARGE AMOUNT</p> <p>S:RUNTYPE="F" LICHARGA=\$\$GETFLICA ; FINAL</p> <p>LINE ITEM CHARGE AMOUNT</p> <p>S:RUNTYPE="F" LIPPA="0.00" ; 13.ACK/PEND</p> <p>LINE ITEM PROV. PAYMENT AMOUNT</p> <p>S:RUNTYPE="F" LIPPA=\$\$GETFLIPPA ; FINAL LINE</p> <p>ITEM PROV. PAYMENT AMOUNT</p> <p>S RVNUCODE=\$P(^CHMXCLF(CHCLFI,1),^",1) ; 14.REVENUE CODE</p> <p>S:RVNUCODE="9999" RVNUCODE="" ; DON'T</p> <p>OUTPUT 9999 VALUE</p> <p>S:RUNTYPE="F" QTYUOS=\$P(CHCLF1,^",8) ; 15.ACK/PEND QUANTITY(UNITS OF SERVICE)</p> <p>S:RUNTYPE="F" QTYUOS=\$\$GETFQTYUOS ; FINAL</p> <p>QUANTITY(UNITS OF SERVICE)</p> <p>S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),^",5) ; 16.EMDEON CLAIM NUMBER</p> <p>S CLMTYP=\$\$GETCLMTYP ; GET</p> <p>I/P/D CLAIM TYPE</p> <p>S:CLMTYP="P" SVCDATES=\$\$GPDSTARTEND ; 17.</p> <p>PROFESSIONAL SERVICE START/END DATE</p> <p>S:CLMTYP="I" SVCDATES=\$\$GIDSTARTEND ; 17.1</p> <p>INSTITUTIONAL SERVICE START/END DATES</p> <p>S:CLMTYP="D" SVCDATES=\$\$GDDSTARTEND ; 17.2 DENTAL</p> <p>SERVICE START/END DATES</p> <p>S SVCSTDATE=\$P(SVCDATES,"*",1),SVCENDATE=\$P(SVCDATES,"*",2)</p> <p>Q</p>	

### 3.2.2.6 Check/EFT Date for Split Submission for Secondary Claims

#### Functional Requirement: 2.8.2.6 Check/EFT Date for Split Submission – Secondary Claims

The CP&E system shall use the most recent Check/EFT Date in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split

into multiple claims within the CP&E system and at least one of the claims contains a Check/EFT Date.

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.6	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman Process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:'\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 </pre>		

Routine Name	CHMXWB21
	<pre> .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK          ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)=" " S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI=" " D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE                                ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)=" " .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY=" " D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,'D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^1,"^",4),CHKEFT=\$P(^1,"^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^0,"^",10) .I CHKDATE'=" " N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE'=" " N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q  GETSTYPE(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S:PDI=" " PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) I PDI I \$D(^CHMPAY("C",PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY,OK S OK=0 I \$O(^CHMXCLE(CHCLEI,80,"B",0)) D .S CHMPAY=0,OK=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY=" " D Q:'OK .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 Q .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=0 Q .. S OK=0 .. Q Q OK  CLM ; </pre>

Routine Name	CHMXWB21
N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDY,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D .I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR) .E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR) W REC,! S REC="",COUNT=COUNT+1 Q	

### 3.2.2.7 Check/EFT Number for Split Submission for Secondary Claims

#### Functional Requirement: 2.8.2.7 Check/EFT Number for Split Submission – Secondary Claims

The CP&E system shall use the most recent Check/EFT Number in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system and at least one of the claims contains a Check/EFT Number.

Routine Name	CHMXWB21	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.7	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	

Routine Name	CHMXWB21
Input Attribute Name and Definition	Name: Definition:
Output Attribute Name and Definition	Name: Definition:
Current Logic	
Same as 3.2.1.1	
Modified Logic (Changes are in bold and highlighted)	
<pre> BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS ; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE ; STVAL STATUS VALUE AFTER FROMNT END EDITS N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE THIS .D NOW^%DTC .S FMDATE=% .S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L") .S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L") .S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN TO SECOND S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1) S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. </pre>	

Routine Name	CHMXWB21
<pre> I \$P(\$G(^CHMPAY(CHMPAY,1)), "^", 4) &gt; CHKDATE S CHKDATE = \$P(^ (1), "^", 4), CHKEFT = \$P(^ (1), "^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)), "^", 10) &gt; PAYDATE S PAYDATE = \$P(^ (0), "^", 10) . I CHKDATE = "" N X S X = CHKDATE D H % DTC S CHKDATE = \$ZD(%H, 4) . I PAYDATE = "" N X S X = PAYDATE D H % DTC S CHKDATE = \$ZD(%H, 4) . D CLM . D BLDSTC("CLM") . D LINE Q  GETSType(CHCLEI) ; TYPE OF CLAIM STATUS FOR BLDSTC() N PDI I \$\$FINAL(CHCLEI) Q "F" S PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 2) ; IF PDI ASSIGNED, CHECK FOR "PENDING" S: PDI = "" PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 4) I PDI I \$D(^CHMPAY("C", PDI)) Q "P" Q "A" ; NO PDI, MUST BE "ACK"  FINAL(CHCLEI) ; FINAL STATUS N CHMPAY, OK S OK = 0 I \$O(^CHMXCLE(CHCLEI, 80, "B", 0)) D . S CHMPAY = 0, OK = 1 . F S CHMPAY = \$O(^CHMXCLE(CHCLEI, 80, "B", CHMPAY)) Q: CHMPAY = "" D Q: 'OK .. I \$P(\$G(^CHMPAY(CHMPAY, 0)), "^", 2) = 4 Q .. I \$P(\$G(^CHMPAY(CHMPAY, 0)), "^", 2) = 0 Q .. S OK = 0 .. Q Q OK  CLM ; N BPFTI, BPPN, BPNID, BPLNAME, BPFNAME, BPMNAME, PNAMEX, SPFTID N SPPN, SPNID, SPLNAME, SPFNAME, PMNAME, PNAMEX, EMPIDNUM, EMPNAME, SUBSCID, SUBLNAME N SUBFNAME, SUBMNAME, SUBNAMEX, PATID, PATLNAME, PATFNAME, PATMNAME, PATN AMEX, PATDOB N PATGENDR, ECLMNUM, CLMCHRG, CLMPMT, CAPD, CHKEFTDATE, CHKEFTNUM, BTYPE, P CIDNUM N PATACT, PRSCNUM, VOUCHID, LOCSYSID, GRPNUM, CLMSTDT, CLMENDT N PDI, CHMPAYI, REC, LN, STR ; DO ACK AND PENDING S PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 2) ; IF PDI ASSIGNED, MUST BE "PENDING" S: PDI = "" PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 4) S: PDI CHMPAYI = \$O(^CHMPAY("C", PDI, 0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC = "" ; BUILD THE RECORD F LN = 1:1 S STR = \$T(EMDEONCLM + LN) Q: STR["END OF RECORD" D . I LN = 1 S REC = REC_ \$\$FORMATDATA^CHMXWBUT(STR) . E S REC = REC_ " " _ \$\$FORMATDATA^CHMXWBUT(STR) W REC, ! S REC = "", COUNT = COUNT + 1 Q </pre>	



### 3.2.2.8 Claim Adjudication/Payment Date for Denied Split Claim for Secondary Claims

#### Functional Requirement: 2.8.2.8 Claim Adjudication/Payment Date for Denied Split Submission – Secondary Claims

The CP&E system shall use the most recent Adjudication/Payment Date in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system and payment is denied for all claims.

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.8	
Related Options		
Related Routines	Routines "Called By"	Routines "Called"
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)	<p><b>BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS</b>  <b>; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE</b>  <b>; STVAL STATUS VALUE AFTER FROMNT END EDITS</b>  <b>N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE</b>  <b>I '\$D(ATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE</b>  <b>THIS</b>  <b>.D NOW^%DTC</b>  <b>.S FMDATE=%</b>  <b>.S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")</b>  <b>.S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")</b>  <b>.S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN</b>  <b>TO SECOND</b>  <b>S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES</b>  <b>S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES</b>  <b>S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1)</b></p>	

Routine Name	CHMXWB21
	<pre> S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1) S:\$D(COUNT) COUNT=0 K CHRJARR S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45 .D CLM .D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS I STYPE="A" D .D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR) .I \$G(STVAL)="" S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT END EDIT FAIL) .I STVAL=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET A2/20 .D CLM .D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS .K CHRJARR S CHRJARR(0)=0 .S CHCLFI=0 .F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI="" D ..D GLINRJRSN^CHMXWBUT(CHCLFI) .D:CHRJARR(0)&gt;0 LINE ; NO LINE REJECTS RECORDED = NO LINE REPORTS I STYPE="F" D .S (CHKDATE,CHKEFT,PAYDATE)="" .S CHRJARR(0)=1 .F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY="" D .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65*" .. I \$P(\$G(^CHMPAY(HACCLM,0)),"^",2)=0,\$D(CHRJARR(1,1)) S CHRJARR(1,1)="F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)),"^",4)&gt;CHKDATE S CHKDATE=\$P(^("1","^",4),CHKEFT=\$P(^("1","^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",10)&gt;PAYDATE S PAYDATE=\$P(^("0","^",10) .I CHKDATE="" N X S X=CHKDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .I PAYDATE="" N X S X=PAYDATE D H^%DTC S CHKDATE=\$ZD(%H,4) .D CLM .D BLDSTC("CLM") .D LINE Q  CLM ; N BPFTI,BPPN,BPNID,BPLNAME,BPFNAME,BPMNAME,PNAMEX,SPFTID N SPPN,SPNID,SPLNAME,SPFNAME,PMNAME,PNAMEX,EMPIDNUM,EMPNAME,SUBSCID, SUBLNAME N SUBFNAME,SUBMNAME,SUBNAMEX,PATID,PATLNAME,PATFNAME,PATMNAME,PATN AMEX,PATDOB N PATGENDR,ECLMNUM,CLMCHRG,CLMPMT,CAPD,CHKEFTDATE,CHKEFTNUM,BTYPE,P CIDNUM N PATACT,PRSCNUM,VOUCHID,LOCSYSID,GRPNUM,CLMSTDT,CLMENDT N PDI,CHMPAYI,REC,LN,STR ; DO ACK AND PENDING S PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",2) ; IF PDI ASSIGNED, MUST BE "PENDING" S:PDI="" PDI=\$P(\$G(^CHMXCLE(CHCLEI,100)),"^",4) S:PDI CHMPAYI=\$O(^CHMPAY("C",PDI,0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC="" ; BUILD THE RECORD </pre>

Routine Name	CHMXWB21
	<p><b>F LN=1:1 S STR=\$T(EMDEONCLM+LN) Q:STR["END OF RECORD" D</b>  <b>.I LN=1 S REC=REC_\$\$FORMATDATA^CHMXWBUT(STR)</b>  <b>.E S REC=REC_" "_\$\$FORMATDATA^CHMXWBUT(STR)</b>  <b>W REC,! S REC="",COUNT=COUNT+1</b>  <b>Q</b></p> <p>GETCLMVAL  N CHCLB0,CHCLB1,CHMPAY1,CHCLE60  N TMPDTE S TMPDTE=0  S CHCLB0=\$G(^CHMXCLB(CHCLBI,0))  S CHCLB1=\$G(^CHMXCLB(CHCLBI,1))  S CHCLE64=\$G(^CHMXCLE(CHCLEI,64))  <b>S:RUNTYPE="F" CHMPAY1=\$G(^CHMPAY(CHMPAYI,1))</b>  S BPFTI=\$P(CHCLB0,"^",2) ; 5.BILLING PROVIDER FEDERAL TAX ID  S BPPN=""  ; 6.BILLING PROVIDER PAYER NUMBER  S BPNID=\$P(CHCLB0,"^",13) ; 7.BILLING PROVIDER NATIONAL ID  S BPLNAME=\$P(CHCLB0,"^",3) ; 8.BILLING PROVIDER LAST NAME  S BPFNAME="" ;  9.BILLING PROVIDER FIRST NAME  S BPMNAME=""  ; 10.BILLING PROVIDER MIDDLE NAME  S BPNAMEX="" ;  11.BILLING PROVIDER NAME SUFFIX  S SPFTID=""  ; 12.RENDERING PHYSICIAN FEDERAL TAX ID  S SPPN=""  ; 13.RENDERING PHYSICIAN PAYER NUMBER  S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID  S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME  S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME  S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME  S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX  S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER  S EMPNAME="" ; 20.EMPLOYER NAME  S CHCLC0=\$G(^CHMXCLC(CHCLCI,0))  S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID  S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME  S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME  S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME  S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX  S PATID="" ; 26.PATIENT ID  S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME  S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME  S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME  S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX  S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH  S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER  S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN  S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER  S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)&gt;0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1))  ; 34.CHARGE AMT  <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b>  <b>AMOUNT</b>  <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b></p>

Routine Name	CHMXWB21
<b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$S(RUNTYPE="F":",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)),^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACCT=\$P(\$G(^CHMXCLE(CHCLEI,0)),^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER S VOUCHID="" ; 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47. PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1 INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2 DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q	

### 3.2.2.9 Claim Adjudication/Payment Date for Paid Split Claim for Secondary Claims

#### Functional Requirement: 2.8.2.9 Claim Adjudication/Payment Date for Paid Split Submission – Secondary Claims

The CP&E system shall use the Adjudication/Payment Date from the most recent paid claim in the Claim Level Record of the Final Claim Status file for an electronic claim submission that is split into multiple claims within the CP&E system.

Routine Name	CHMXWB21	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.2.9	
Related Options		
Related Routines	Routines “Called By”	Routines “Called”
	Taskman process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		

Routine Name	CHMXWB21
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local
Input Attribute Name and Definition	Name: Definition:
Output Attribute Name and Definition	Name: Definition:
Current Logic	
Same as 3.2.1.1	
Modified Logic (Changes are in bold and highlighted)	<p><b>BLDCLM(CHCLEI,STVAL) ;BUILD UP THE CLAIM RECORDS</b>  <b>; CHCLEI CLAIM NUMBER FROM INCOMING 837 FILE</b>  <b>; STVAL STATUS VALUE AFTER FROMNT END EDITS</b>  <b>N CHCLCI,CHCLBI,CHCLAI,CHCLI,STYPE,TMP,CHKDATE,CHKEFT,CHMPAY,PAYDATE</b>  <b>I '\$D(DATESTAMP) D ; PROVIDED FOR TESTING, PRODUCTION SHOULD NOT REQUIRE</b>  <b>THIS</b>  <b>.D NOW^%DTC</b>  <b>.S FMDATE=%</b>  <b>.S FMDATE=\$\$JUSTIFY^CHMXWBUT(FMDATE,14,0,"L")</b>  <b>.S TIME=\$\$JUSTIFY^CHMXWBUT(\$E(FMDATE,9,14),6,0,"L")</b>  <b>.S DATESTAMP=(\$E(FMDATE,1,7)+17000000)_TIME ; DATE/TIME FILE CREATED DOWN</b>  <b>TO SECOND</b>  <b>S CHCLCI=\$P(^CHMXCLE(CHCLEI,0),"^",1) ;TRAVERSE BACK THROUGH BUFFER FILES</b>  <b>S CHCLBI=\$P(^CHMXCLC(CHCLCI,0),"^",1) ;GET "I" VALUES FOR ALL BUFFER FILES</b>  <b>S CHCLAI=\$P(^CHMXCLB(CHCLBI,0),"^",1)</b>  <b>S CHCLI=\$P(^CHMXCLA(CHCLAI,0),"^",1)</b>  <b>S:'\$D(COUNT) COUNT=0</b>  <b>K CHRJARR</b>  <b>S STYPE=\$\$GETSTYPE(CHCLEI) ; "TYPE" PENDING/ACK/FINAL</b>  <b>I STYPE="P" S CHRJARR(0)=1 S CHRJARR(1,1)="P1*45*" D ; ALL "PENDING" = P1/45</b>  <b>.D CLM</b>  <b>.D BLDSTC("CLM") ; EACH CLAIM GETS CLM, STC RECORDS</b>  <b>I STYPE="A" D</b>  <b>.D BLDACK ; BUILD "ACK" REJECT REASON CODES ARRAY (CHRJARR)</b>  <b>.I \$G(STVAL)=" " S STVAL=\$\$GETSTVAL(CHCLEI) ; GET STATUS OF CLAIM (6=FRONT</b>  <b>END EDIT FAIL)</b>  <b>.I STVAL'=6 S CHRJARR(0)=1,CHRJARR(1,1)="A2*20*" ; NO FRONT END EDITS GET</b>  <b>A2/20</b>  <b>.D CLM</b>  <b>.D BLDSTC("CLM") ; ALL "ACKS" GET A STATUS</b>  <b>.K CHRJARR S CHRJARR(0)=0</b>  <b>.S CHCLFI=0</b>  <b>.F S CHCLFI=\$O(^CHMXCLF("B",CHCLEI,CHCLFI)) Q:CHCLFI=" " D</b>  <b>..D GLINRJRSN^CHMXWBUT(CHCLFI)</b>  <b>.D:CHRJARR(0)&gt;0 LINE ; NO LINE</b>  <b>REJECTS RECORDED = NO LINE REPORTS</b>  <b>I STYPE="F" D</b>  <b>. S (CHKDATE,CHKEFT,PAYDATE)=" "</b>  <b>. S CHRJARR(0)=1</b>  <b>. F S CHMPAY=\$O(^CHMXCLE(CHCLEI,80,"B",CHMPAY)) Q:CHMPAY=" " D</b>  <b>.. I \$P(\$G(^CHMPAY(CHMPAY,0)),"^",2)=4 S CHRJARR(1,1)="F1*65"</b></p>

Routine Name	CHMXWB21
<pre> .. I \$P(\$G(^CHMPAY(HACCLM,0)), "^", 2) = 0, \$D(CHRJARR(1,1)) S CHRJARR(1,1) = "F2*1*" .. I \$P(\$G(^CHMPAY(CHMPAY,1)), "^", 4) &gt; CHKDATE S CHKDATE = \$P(^1, "^", 4), CHKEFT = \$P(^1, "^", 16) .. I \$P(\$G(^CHMPAY(CHMPAY,0)), "^", 10) &gt; PAYDATE S PAYDATE = \$P(^0, "^", 10) . I CHKDATE = "" N X S X = CHKDATE D H^%DTC S CHKDATE = \$ZD(%H, 4) . I PAYDATE = "" N X S X = PAYDATE D H^%DTC S CHKDATE = \$ZD(%H, 4) . D CLM . D BLDSTC("CLM") . D LINE Q  CLM ; N BPFTI, BPPN, BPNID, BPLNAME, BPFNAME, BPMNAME, PNAMEX, SPFTID N SPPN, SPNID, SPLNAME, SPFNAME, PMNAME, PNAMEX, EMPIDNUM, EMPNAME, SUBSCID, SUBLNAME N SUBFNAME, SUBMNAME, SUBNAMEX, PATID, PATLNAME, PATFNAME, PATMNAME, PATN AMEX, PATDOB N PATGENDR, ECLMNUM, CLMCHRG, CLMPMT, CAPD, CHKEFTDATE, CHKEFTNUM, BTYPE, P CIDNUM N PATACT, PRSCNUM, VOUCHID, LOCSYSID, GRPNUM, CLMSTDT, CLMENDT N PDI, CHMPAYI, REC, LN, STR ; DO ACK AND PENDING S PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 2) ; IF PDI ASSIGNED, MUST BE "PENDING" S: PDI = "" PDI = \$P(\$G(^CHMXCLE(CHCLEI, 100)), "^", 4) S: PDI CHMPAYI = \$O(^CHMPAY("C", PDI, 0)) D GETCLMVAL ; SET UP THE CLAIM VALUE VARIABLES S REC = "" ; BUILD THE RECORD F LN = 1:1 S STR = \$T(EMDEONCLM+LN) Q: STR["END OF RECORD" D . I LN = 1 S REC = REC_ \$\$FORMATDATA^CHMXWBUT(STR) . E S REC = REC_ " " _ \$\$FORMATDATA^CHMXWBUT(STR) W REC, ! S REC = "", COUNT = COUNT + 1 Q  GETCLMVAL N CHCLB0, CHCLB1, CHMPAY1, CHCLE60 N TMPDTE S TMPDTE = 0 S CHCLB0 = \$G(^CHMXCLB(CHCLBI, 0)) S CHCLB1 = \$G(^CHMXCLB(CHCLBI, 1)) S CHCLE64 = \$G(^CHMXCLE(CHCLEI, 64)) S: RUNTYPE = "F" CHMPAY1 = \$G(^CHMPAY(CHMPAYI, 1)) S BPFTI = \$P(CHCLB0, "^", 2) ; 5.BILLING PROVIDER FEDERAL TAX ID S BPPN = "" ; 6.BILLING PROVIDER PAYER NUMBER S BPNID = \$P(CHCLB0, "^", 13) ; 7.BILLING PROVIDER NATIONAL ID S BPLNAME = \$P(CHCLB0, "^", 3) ; 8.BILLING PROVIDER LAST NAME S BPFNAME = "" ; 9.BILLING PROVIDER FIRST NAME S BPMNAME = "" ; 10.BILLING PROVIDER MIDDLE NAME S BPNAMEX = "" ; 11.BILLING PROVIDER NAME SUFFIX S SPFTID = "" ; 12.RENDERING PHYSICIAN FEDERAL TAX ID </pre>	

Routine Name	CHMXWB21
S SPPN="" ; 13.RENDERING PHYSICIAN PAYER NUMBER S SPNID=\$P(CHCLE64,"^",4) ; 14.RENDERING PHYSICIAN NATIONAL ID S SPLNAME=\$P(CHCLE64,"^",1) ; 15.RENDERING PHYSICIAN NAME S SPFNAME=\$P(CHCLE64,"^",2) ; 16.RENDERING PHYSICIAN FNAME S SPMNAME=\$P(CHCLE64,"^",3) ; 17.RENDERING PHYSICIAN MNAME S SPNAMEX="" ; 18.RENDERING PHYSICIAN NAME SUFFIX S EMPIDNUM="" ; 19.EMPLOYER IDENTIFICATION NUMBER S EMPNAME="" ; 20.EMPLOYER NAME S CHCLC0=\$G(^CHMXCLC(CHCLCI,0)) S SUBSCID=\$P(CHCLC0,"^",4) ; 21.SUBSCRIBER ID S SUBLNAME=\$P(CHCLC0,"^",5) ; 22.SUBSCRIBER LAST NAME S SUBFNAME=\$P(CHCLC0,"^",6) ; 23.SUBSCRIBER FIRST NAME S SUBMNAME=\$P(CHCLC0,"^",7) ; 24.SUBSCRIBER MIDDLE NAME S SUBNAMEX=\$P(CHCLC0,"^",8) ; 25.SUBSCRIBER NAME SUFFIX S PATID="" ; 26.PATIENT ID S PATLNAME=\$P(CHCLC0,"^",5) ; 27.PATIENT LAST NAME S PATFNAME=\$P(CHCLC0,"^",6) ; 28.PATIENT FIRST NAME S PATMNAME=\$P(CHCLC0,"^",7) ; 29.PATIENT MIDDLE NAME S PATNAMEX=\$P(CHCLC0,"^",8) ; 30.PATIENT NAME SUFFIX S PATDOB=\$P(CHCLC0,"^",9) ; 31.PATIENT DATE OF BIRTH S PATGENDR=\$P(CHCLC0,"^",10) ; 32.PATIENT GENDER S:"MF"[PATGENDR PATGENDR="" ; DO NOT ALLOW PATIENT GENDER UNKNOWN S ECLMNUM=\$P(^CHMXCLE(CHCLEI,3),"^",5) ; 33.EMDEON CLAIM NUMBER S CLMCHRG=\$S(\$P(^CHMXCLE(CHCLEI,2),"^",1)>0:"0.00",1:\$P(^CHMXCLE(CHCLEI,2),"^",1)) ; 34.CHARGE AMT <b>S CLMPMT=\$S(RUNTYPE="F":\$P(CHMPAY1,"^",1),1:"0.00") ; 35.CLAIM PAYMENT</b> <b>AMOUNT</b> <b>S CAPD=\$S(RUNTYPE="F":PAYDATE,1:"") ; 36.CLAIM ADJ/PAYMENT DATE</b> <b>S CHKEFTDATE=\$S(RUNTYPE="F":CHKDATE,1:"") ; 37.CHECK/EFT DATE</b> <b>S CHKEFTNUM=\$S(RUNTYPE="F":CHKEFT,1:"") ; 38.CHECK/EFT NUMBER</b> S BTYPE=\$S(RUNTYPE="F":",",1:"") ; 39.BILL TYPE S PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",2) ; 40.PAYER CLAIM ID NUMBER (PDI) S:PCIDNUM="" PCIDNUM=\$P(\$G(^CHMXCLE(CHCLEI,100)), "^",4) ; PDI NOT POPULATED, DERIVED CLAIM CTL NUM S PATACT=\$P(\$G(^CHMXCLE(CHCLEI,0)), "^",2) ; 41.PATIENT ACCOUNT NUMBER S PRSCNUM="" ; 42.PHARMACY PRESCRIPTION NUMBER S VOUCHID="" ; 43.VOUCHER IDENTIFIER S LOCSYSID="" ; 44.APP/LOCATION SYSTEM ID S GRPNUM="" ; 45.GROUP NUMBER S CLMTYP=\$GETCLMTYP ; GET I/P/D CLAIM TYPE S:CLMTYP="P" SVCDATES=\$\$GPSTARTEND ; 46/47. PROFESSIONAL SERVICE START DATE S:CLMTYP="I" SVCDATES=\$\$GISTARTEND ; 46.1/47.1 INSTITUTIONAL START/END S:CLMTYP="D" SVCDATES=\$\$GDSTARTEND ; 46.2/47.2 DENTAL START/END S CLMSTDT=\$P(SVCDATES,"*",1),CLMENDT=\$P(SVCDATES,"*",2) Q	

### 3.2.3 Final Status One-Time Update for Primary Claims for CHAMPVA/SB/CWVV

#### 3.2.3.1 Create a Final Status Update for Historical Primary Claims

##### Functional Requirement: 2.8.3.1 Final Status Update – Primary Claims

The CP&E system shall FTP a one-time Final Status file(s) for existing primary claims that are in a pending status that should be in a final status. All claims after the SLLA project was released to production and claims were processed shall be included in this one-time historical update, or at most 6 months worth of data. The date which the SLLA software began processing claims was 7/27/13. When the process is initially run at the time of implementation, all claims that have been completed within the last six (6) months or since 7/27/13, whichever is the latest date, will be included in the initial Final Status file(s) created and transmitted to Emdeon. Subsequent nightly execution of this process will only include claims that have gone final since the prior running of this process.

Routine Name	<b>CHMXWB21</b>	
Enhancement Category	<input type="checkbox"/> New <input checked="" type="checkbox"/> Modify <input type="checkbox"/> Delete <input type="checkbox"/> No Change	
Requirement Traceability Matrix	2.8.3.1	
Related Options		
Related Routines	Routines “Called By”	Routines “Called”
	TaskMan process	CHMXWBUT
Data Dictionary (DD) References		
Related Protocols		
Related Integration Control Registrations (ICRs)		
Data Passing	<input type="checkbox"/> Input <input type="checkbox"/> Output Reference <input type="checkbox"/> Both <input type="checkbox"/> Global Reference <input type="checkbox"/> Local	
Input Attribute Name and Definition	Name: Definition:	
Output Attribute Name and Definition	Name: Definition:	
Current Logic		
Same as 3.2.1.1		
Modified Logic (Changes are in bold and highlighted)		



Routine Name	CHMXWB21
<pre> EPFINAL      ; entry point to start Final Status Creation file N RUNTYPE,GRPCNT,EICNT,GROUPID,CHTPI,CHTPARR,CHFROM,CHTO,FILECNT N TRKI N COUNT,DIRFILE,SEQCNT,LSTTIME,HOLD,DATESTAMP,STVAL,FMDATE N CHNODEDEF,BREF,CHIJKVAL,DRDATA,TRKDATA,MAXCNT      ; FILEMAN UPDATE VARIABLES S RUNTYPE="F",GRPCNT=1,EICNT=0 ; INITIALIZE FROM/TO INDICES,STATUS TYPE D INIT S CHTPI="A" F S CHTPI=\$O(^CHMX277("B",CHTPI),-1) Q:CHTPI="" D      ;TRADING PARTNERS FROM "B" XREF .K CHTPARR S (CHFROM,CHTO,FILECNT,CHMEDCOB)=0 ; KILL TRADING PARTNER ARRAY; ASSUME NOT A "COB" TRADING PARTNER .S TRKI=\$O(^CHMX277("B",CHTPI,0)) ; GET TRACKING GLOBAL INDEX .I \$P(\$G(^CHMX277(TRKI,0)),"^",3)=0 Q .I \$P(^CHMX277(TRKI,0),"^",4)["COB" S CHMEDCOB=1 ; TRADING PARTNER 11 = "COB" EMDEON PARTNER .S CHTPARR(CHTPI)=" " .D GETTOFROM(CHTPI) Q:(^CHFROM) ; RETRIEVE FROM/TO FROM TRACKING GLOBAL .D GETFINALEI(CHFROM,CHTO) ; GENERATE ^ZSC ARRAY OF ^CHMXCLE(I) VALUES .Q:^ZSC(\$J,0)=0 ; NO RECORDS FOUND, CREATE NO FILE .D CREATEFILE      ; CREATE THE PENDING STATUS FILE .D FMUPDATE(CHTPI) Q  GETFINALEI(CHFROM,CHTO) ; GET CHCLEI FOR FINAL CLAIM STATUS K ^ZSC(\$J),^ZSC1(\$J) N CHCLEI,CHCLAI,CTPI,RECCNT,CLC,CLB,CHPIDX,CHPDATE,CHPPDI S RECCNT=0 S %H=\$H-180 D YMD^%DTC S CHPDATE=X F S CHPDATE=\$O(^CHMPAY("E",CHPDATE)) Q:CHPDATE="" S CHPIDX="" F S CHPIDX=\$O(^CHMPAY("E",CHPDATE,CHPIDX)) Q:CHPIDX="" I \$G(^CHPIDX))'=1 D .S CHPPDI=\$O(^CHMPAY(CHPIDX,"PDI",0)) Q:CHPPDI="" .I CHPPDI=\$P(\$G(^CHMPAY(CHPIDX,"PDI",PDI,0)),"^") .I CHPPDI="" Q .S CHCLEI=\$O(^CHMXCLE("D",CHPPDI,"")) I CHCLEI="" Q .I \$\$FINAL(CHCLEI) D .. S CLC=\$P(^CHMXCLE(CLE,0),"^") .. S CLB=\$P(^CHMXCLC(CLC,0),"^") .. S CHCLAI=\$P(^CHMXCLB(CLB,0),"^") .. S CTPI=\$\$GETTP(CHCLAI) ; TRADING PARTNER FOR CURRENT CLAIM .. I \$D(CHTPARR(CTPI)) S ^ZSC(\$J,CTPI,CHCLEI)="",RECCNT=RECCNT+1,^ZSC1(\$J,CHPDATE,CHPIDX)="" S ^ZSC(\$J,0)=RECCNT Q  FMUPDATE(CHTPI) ; UPDATE THE TRACKING GLOBAL USING FILEMAN Q:"APF"[RUNTYPE      ; A=ACK, P=PENDING, F=FINAL N CHNODEDEF,BREF,CHIJKVAL,DRDATA,TRKDATA S:RUNTYPE="A" CHNODEDEF="^CHMX277(I,10,"TRKDATA=".02=1^.03="_DIRFILE_".^04="_COUN </pre>	

Routine Name	CHMXWB21
	<pre> T_".05=" _CHMXI   S:RUNTYPE="P" CHNODEDEF="^CHMX277(I,20," ,TRKDATA=".02=1^.03=" _DIRFILE_ ".04=" _CHTO _ ".05=" _CHFROM_ ".06=" _COUNT   S:RUNTYPE="F" CHNODEDEF="^CHMX277(I,30," ,TRKDATA=".02=1^.03=" _DIRFILE_ ".04=" _CHTO _ ".05=" _CHFROM_ ".06=" _COUNT   S BREF=FMDATE   S CHIJKVAL=\$O(^CHMX277("B",CHTPI,0)) ; ADD REQUIRES "I", NO "J" VALUE     S DRDATA=\$\$SETDARR(TRKDATA) ; SET REG/VALUE FOR FILEMAN FORMAT     D ADD^CHHRLIBFM(CHNODEDEF,CHIJKVAL,BREF,DRDATA) ; FILEMAN UPDATE FOR TRACKING GLOBAL       I RUNTYPE="F" N X,X1 F S X=\$O(^ZSC1(\$J,X)) Q:X="" F S X1=\$O(^ZSC1(\$J,X,X1)) Q:X1="" I \$D(^CHMPAY("E",X,X1)) S ^CHMPAY("E",X,X1)=1 Q </pre>

## Attachment A - Approval Signatures

This section is used to document the approval of the Software Design Document during the Formal Review. The review should be ideally conducted face to face where signatures can be obtained 'live' during the review however the following forms of approval are acceptable:

1. Physical signatures obtained face to face or via fax
2. Digital signatures tied cryptographically to the signer
3. /es/ in the signature block provided that a separate digitally signed e-mail indicating the signer's approval is provided and kept with the document

---

Signed:	Date:
< <i>Integrated Project Team (IPT) Chair</i> >	

---

Signed:	Date:
< <i>Business Sponsor</i> >	

---

Signed:	Date:
< <i>IT Program Manager</i> >	

---

Signed:	Date:
< <i>Project Manager</i> >	

---

Signed:	Date:
< <i>Technical and Enterprise Architectural Review Team</i> >	