**Medical Care Collection Fund (MCCF)**

**eInsurance Compliance Phase 3**

**NSR #20140413**

System Design Document

IB\*2.0\*549



Department of Veterans Affairs

**May 2015**

Version 2.0

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 05/13/2015 | 2.0 | Applied VA feedback to the document. |  |
| 04/24/2015 | 1.0 | Initial Document |  |

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product. Per the Project Management Accountability System (PMAS) Guide, the SDD as a conceptual design is required prior to the Milestone 1 Review. (Sections 1, 2, 3, 4, 5, 7, 9 need to be populated, as applicable.) The as-built design for each delivery must be incorporated prior to the Milestone 2 Review. (The entire document needs to be populated or updated, as applicable.)

Table of Contents

[1 Introduction 1](#_Toc419281093)

[1.1 Purpose of the SDD 1](#_Toc419281094)

[1.2 Identification 1](#_Toc419281095)

[1.3 Scope 2](#_Toc419281096)

[1.4 Constraining Policies, Directives and Procedures 3](#_Toc419281097)

[1.5 User Characteristics 4](#_Toc419281098)

[1.6 Relationship to Other Documents and Plans 4](#_Toc419281099)

[1.7 Definitions, Acronyms, and Abbreviations 4](#_Toc419281100)

[1.8 References 4](#_Toc419281101)

[2 Background 6](#_Toc419281102)

[2.1 Overview of the System 6](#_Toc419281103)

[2.2 Overview of the Business Process 6](#_Toc419281104)

[2.3 Business Benefits 8](#_Toc419281105)

[2.4 Assumptions and Constraints 9](#_Toc419281106)

[2.4.1 Design Assumptions 9](#_Toc419281107)

[2.4.2 Design Constraints 9](#_Toc419281108)

[2.4.3 Design Trade-offs 9](#_Toc419281109)

[2.5 Overview of the Significant Requirements 9](#_Toc419281110)

[2.5.1 Overview of Significant Functional Requirements 9](#_Toc419281111)

[2.5.2 Overview of Functional Workload / Performance Requirements 11](#_Toc419281112)

[2.5.3 Overview of Operational Requirements 11](#_Toc419281113)

[2.5.4 Overview of the Technical Requirements 11](#_Toc419281114)

[2.5.5 Overview of the Security or Privacy Requirements 11](#_Toc419281115)

[2.5.6 Overview of System Criticality and High Availability Requirements 12](#_Toc419281116)

[2.5.7 Single Sign-on Requirement 12](#_Toc419281117)

[2.5.8 Requirement for Use of Enterprise Portals 12](#_Toc419281118)

[2.5.9 Special Device Requirements 12](#_Toc419281119)

[2.6 Legacy System Retirement 12](#_Toc419281120)

[3 Conceptual Design 13](#_Toc419281121)

[3.1 Conceptual Application Design 13](#_Toc419281122)

[3.1.1 Application Context 13](#_Toc419281123)

[3.1.2 High-Level Application Design 13](#_Toc419281124)

[3.1.3 Application Locations 13](#_Toc419281125)

[3.2 Conceptual Data Design 13](#_Toc419281126)

[3.2.1 Project Conceptual Data Model 13](#_Toc419281127)

[3.2.2 Database Information 13](#_Toc419281128)

[3.2.3 User Interface Data Mapping 13](#_Toc419281129)

[3.3 Conceptual Infrastructure Design 14](#_Toc419281130)

[3.3.1 System Criticality and High Availability 14](#_Toc419281131)

[3.3.2 Special Technology 14](#_Toc419281132)

[3.3.3 Technology Locations 14](#_Toc419281133)

[3.3.4 Conceptual Infrastructure Diagram 14](#_Toc419281134)

[4 System Architecture 15](#_Toc419281135)

[4.1 Hardware Architecture 15](#_Toc419281136)

[4.2 Software Architecture 15](#_Toc419281137)

[4.3 Network Architecture 15](#_Toc419281138)

[4.4 Service Oriented Architecture / ESS 15](#_Toc419281139)

[4.5 Enterprise Architecture 15](#_Toc419281140)

[5 Data Design 16](#_Toc419281141)

[5.1 DBMS Files 16](#_Toc419281142)

[5.2 Non-DBMS Files 21](#_Toc419281143)

[5.3 Data View 21](#_Toc419281144)

[6 Detailed Design 22](#_Toc419281145)

[6.1 Hardware Detailed Design 22](#_Toc419281146)

[6.2 Software Detailed Design 22](#_Toc419281147)

[6.2.1 Conceptual Design 22](#_Toc419281148)

[6.2.2 Specific Requirements 23](#_Toc419281149)

[6.3 Network Detailed Design 277](#_Toc419281150)

[6.4 Service Oriented Architecture / ESS Detailed Design 277](#_Toc419281151)

[6.4.1 Service Description for <Consumed Service Name> 277](#_Toc419281152)

[6.4.2 Service Design for <Provided Service Name> 278](#_Toc419281153)

[7 External System Interface Design 280](#_Toc419281154)

[7.1 Interface Architecture 280](#_Toc419281155)

[7.2 Interface Detailed Design 280](#_Toc419281156)

[8 Human-Machine Interface 281](#_Toc419281157)

[8.1 Interface Design Rules 281](#_Toc419281158)

[8.2 Inputs 281](#_Toc419281159)

[8.3 Outputs 281](#_Toc419281160)

[8.4 Navigation Hierarchy 281](#_Toc419281161)

[8.4.1 Screen [x.1] 281](#_Toc419281162)

[9 Security and Privacy 282](#_Toc419281163)

[9.1 Security 282](#_Toc419281164)

[9.2 Privacy 282](#_Toc419281165)

[Attachment A – Approval Signatures 283](#_Toc419281166)

[A. Additional Information 284](#_Toc419281167)

[A.1. RTM 284](#_Toc419281168)

[A.2. Packaging and Installation 284](#_Toc419281169)

[A.3. Design Metrics 284](#_Toc419281170)

[A.4. Acronym List and Glossary 284](#_Toc419281171)

[A.5. Required Technical Documents 286](#_Toc419281172)

[A.6. Attach Documents 286](#_Toc419281173)

# Introduction

Insurance identification and verification is vital to the success of the Department of Veterans Affairs (VA) revenue collection process. Accurate insurance information is needed to effectively submit claims and collect payments from third-party payers for medical care and services provided to Veterans by the Veterans Health Administration (VHA). In an effort to provide all revenue staff the necessary tools to become more efficient at their collection duties, this patch seeks to provide the software updates and reports needed to assure the insurance information is valid and accessible to all Billing, Accounts Receivable (AR) and Revenue Utilization Review (RUR) staff. The reports will enable Insurance Verification personnel to review and ensure that the information captured in the Patient Insurance File is updated with the latest insurance information possible. Management personnel will be enabled to obtain any insurance information as required by Office of Inspector General (OIG), Chief Business Office (CBO), and Consolidated Patient Account Centers (CPAC) Program Management Office (PMO) in an efficient and timely manner. Standardization is also promoted to ensure insurance information and processes are consistent throughout the nation.

VHA is home to the United States’ largest integrated health care system and is committed to achieving the full benefits of Electronic Data Interchange (EDI) Health Care processes. This project supports VA’s strategic goal to support Veterans’ rights to receive benefits and meet expectations for quality, timeliness, and responsiveness.

This project includes enhancements to VistA ’s Insurance Buffer, electronic Insurance Verification (eIV), Integrated Billing (IB), Insurance Capture Buffer (ICB), and the National Insurance File (NIF) Interface. It also includes enhancements to Patient Policy Comments, Group Policy Comments, and several reports. Finally, this effort includes a new report, changes to the Medical Care Cost Recovery (MCCR) Site Parameters, and documentation which will map 270 and 271 eIV transactions. Furthermore, a trigger is needed on the Patient file (#2) thus requiring a minor change to the Registration (DG) module as well.

## Purpose of the SDD

The purpose of this System Design Document (SDD) is to describe in sufficient detail what VistA modifications are necessary to fulfill the requirements for the Medical Care Collection Fund (MCCF) eInsurance Compliance Phase 3 effort. The SDD translates the requirement specifications into a document from which the developers can create the actual system. It identifies the top-level system architecture, hardware, software, communication, and interface components.

## Identification

The software that this SDD applies to is the Veterans Health Information Systems and Technology Architecture (VistA) IB version 2.0 module and the DG version 5.3 module. The software will use the following standards:

* American National Standards Institute (ANSI)
* International Organization for Standardization (ISO)
* Veterans Administration Standards and Conventions (VA SAC)

For a list of references and links to these documents refer to References (Section 1.8).

For a list of acronyms refer to Acronym List and Glossary (Appendix A.4).

## Scope

Table 1: Scope Inclusions

| Includes |
| --- |
| System Feature: eIV Extract Logic  The ‘eIV Extract Logic’ section of this SDD describes the design associated with the requirements listed in section 2.6.1 of the Requirements Specification Document (RSD). |
| System Feature: Enhance VistA  The ‘Enhance VistA’ section of this SDD describes the design associated with the requirements listed in sections 2.6.2 through 2.6.8 of the RSD.  The following subcategories (which can be found in the RSD) covered by the ‘Enhance VistA’ section of this SDD are as follows.   * Enhance VistA – Date of Death (DOD) * Enhance VistA – Security Key (Key) * Enhance VistA – electronic Insurance Verification (eIV) * Enhance VistA – Integrated Billing (IB) * Enhance VistA – Insurance Capture Buffer (ICB) * Enhance VistA – Health Plan Identifier (HPID) * Enhance VistA – Data Dictionary (DD) |
| System Feature: Insurance Comments  The ‘Insurance Comments’ section of this SDD describes the design associated with the requirements listed in section 2.6.9 of the RSD. |
| System Feature: IV Site Parameters  The ‘IV Site Parameter’ section of this SDD describes the design associated with the requirements listed in section 2.6.10 of the RSD. |
| System Feature: Reports  The ‘Reports’ section of this SDD describes the design associated with the requirements listed in sections 2.6.11 through 2.6.15 of the RSD.  The following subcategories (which can be found in the RSD) covered by the ‘Reports’ section of this SDD are as follows:   * Report – List Plans by Insurance Company * Report – Missing Data (\*New Report) * Report – Patients Without MEDICARE * Report – Active Policies with no Effective Date * Report – eIV Auto Update Report[[1]](#footnote-2) |
| System Feature: Worklist – Move Subscribers to a Different Plan  The ‘Worklist – Move Subscribers to a Different Plan’ section of this SDD describes the design associated with the requirements listed in section 2.6.16 of the RSD. |
| The following requirements involve updating written artifact(s) rather than actual VistA code changes.   |  |  | | --- | --- | | **RSD ID#** | **Requirement Title** | | 2.5.5.1 | Map eIV – Trace eIV’s 270 transmission | | 2.5.5.2 | Map eIV – Document eIV’s 270 transmission | | 2.5.5.3 | Map eIV – Trace eIV’s 271 transmission | | 2.5.5.4 | Map eIV – Document eIV’s 271 transmission | |

Table 2: Scope Exclusion

| Excludes |
| --- |
| Refer to the RSD Appendix C for requirements that have been excluded from the scope of planned work. |

## Constraining Policies, Directives and Procedures

The constraining policies, directives, and procedures that are to be followed by this development effort include VA’s SAC.

## User Characteristics

| **Type of User** | **Description** |
| --- | --- |
| Primary Users | VA Medical Center (VAMC) Patient Registration Teams |
| Primary Insurance Verification Clerks (those employed by facilities and by CPACs) |
| Secondary Insurance Verification Clerks (those employed by facilities and by CPACs) |
| CPAC IV Supervisors, CPAC IV Leads, Facility Revenue Supervisors |
| CPAC IV Managers / Facility Billing Managers |
| CBO CPAC PMO |
| Secondary Users | Veterans Integrated Service Network (VISN) Business Implementation Managers (BIMs) |
| CBO Revenue Operations |
| Business Office Managers/Service Line Managers/Patient Information Collection Management (PICM) |
| CBO eBusiness Solutions Office |
| CBO Business Information Office |
| VAMC Information Resource Managers (IRMs) |
| National Office of Information and Technology (OI&T) |
| National OI&T System Administrators |

## Relationship to Other Documents and Plans

This design document is based off the RSD listed below. The requirements found in the RSD were derived from several user12ence call meetings, emails, a face-to-face three day requirement kick off session, and the Business Requirements Document (BRD) listed below. The location of these documents can be found in References (Section 1.8), of this document.

* **20140413 MCCF eInsurance Compliance Phase 3 BRD**
* **09-029 eInsurance RSD v3.00**

## Definitions, Acronyms, and Abbreviations

Refer to Acronym List and Glossary (Section A.4), of this document.

## References

The following references were used in the development of this SDD:

* **20140413 MCCF eInsurance Compliance Phase 3 BRD**

<http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active>

* **MCCF FY15 eInsurance Compliance Requirements Meeting Minutes**

<http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active>

* **Integrated Billing (IB) V. 2.0 User Manual**

<http://www.domain/vdl/documents/Financial_Admin/Integrated_Billing_(IB)/ib_2_0_um_r0115.doc>

* **HIPAA (Health Insurance Portability and Accountability Act of 1996)**

<http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf>

* **Public Law 111publ148, The Patient Protection and Affordable Care Act**

<http://www.gpo.gov/fdsys/pkg/PLAW-111publ148/pdf/PLAW-111publ148.pdf>

* **PPACA Compliance, Certification, and Penalties**

<https://www.cms.gov/Affordable-Care-Act/04_ComplianceCertificationandPenalties.asp>

* **The Department of Veterans Affairs M Programming Standards and Conventions**

<http://vaww.yourserver.domain/communities/app_dev/sac/default.aspx>

* **09-029 eInsurance Workflow Current v2.00**

[http://your\_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active](http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active%20)

* **09-029 eInsurance RSD v3.00**

The RSD will be posted to the following location upon client’s approval of the document: [http://your\_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active](http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active%20)

* **09-029 eInsurance ERD v1.00 – full page**

The Entity Relationship Diagram (ERD) (full page) will be posted to the following location upon client’s approval of the document: [http://your\_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active](http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active%20)

* **09-029 eInsurance ERD v1.00 – 1 page**

The ERD (1 page) will be posted to the following location upon client’s approval of the document: [http://your\_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active](http://your_srver.domain.ext/warboard/anotebk.asp?proj=1733&Type=Active%20)

# Background

## Overview of the System

The VistA users enter patient insurance information in the Integrated Billing module through a variety of processes:

* Insurance information may be entered manually during the Registration process
* It may be entered when the patient’s insurance card is read by the insurance card reader
* A user may enter patient’s insurance information directly into the Patient file using the Patient Insurance Info View/Edit option
* A user may enter patient’s insurance information into the Insurance Buffer in VistA either using ICB or by using the VistA option Process Insurance Buffer

Regardless of how the patient’s insurance information gets entered into VistA, it must be verified with the insurance company and the verification must be periodically updated. The goal of the eIV process is to automate as much of the verification process as possible to ensure that the insurance information, used to submit claims for services rendered to the patient, is accurate and up-to-date. This in turn increases the likelihood of timely reimbursement and increased revenue.

## Overview of the Business Process

The diagram below shows a high-level overview of the eIV process from the Veteran Affairs Medical Centers (VAMCs) to the third-party payers, including intermediate systems.

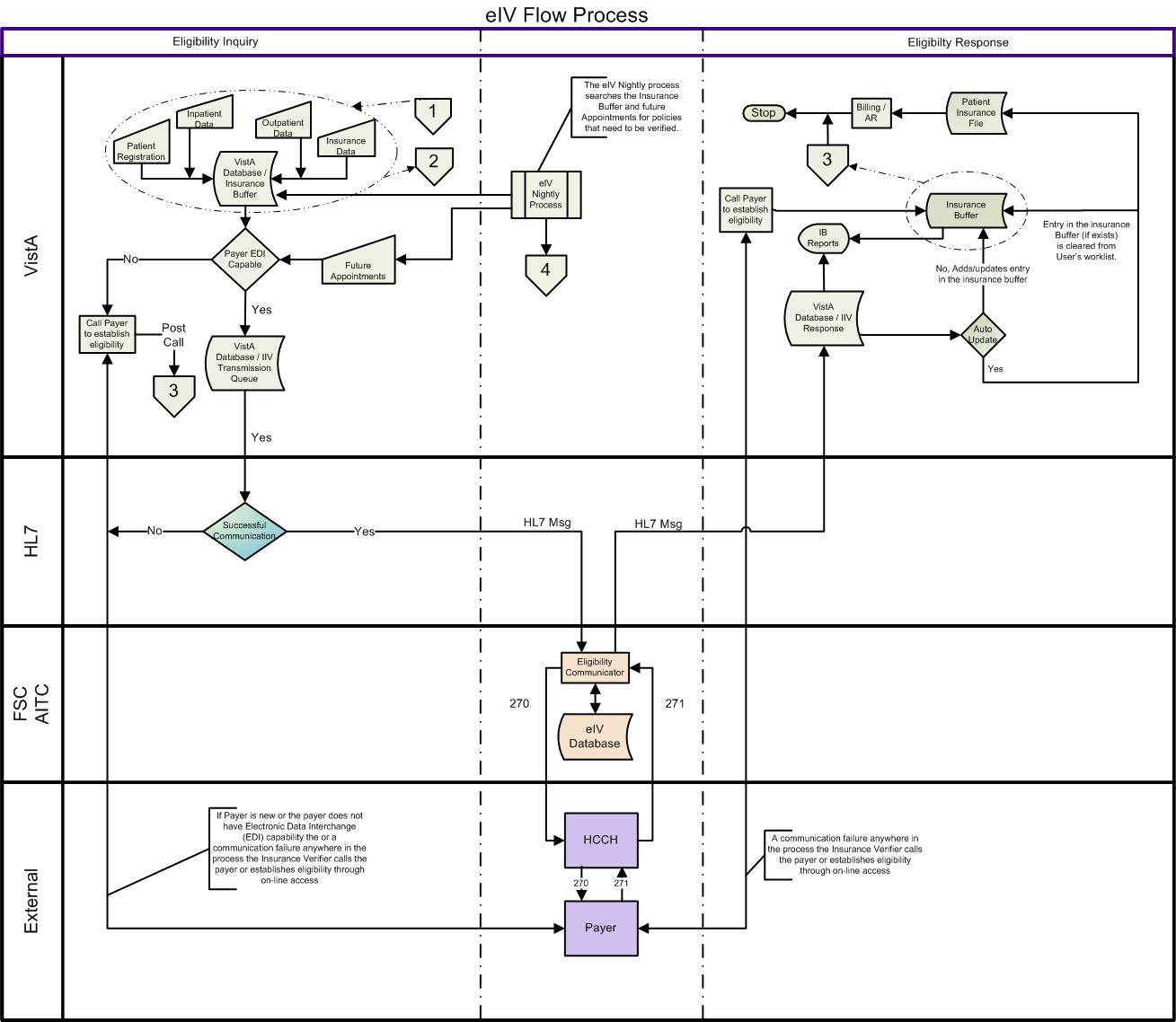


Figure 1: eIV Process High-Level Overview

#1

Sources of information that populate the VistA Database/Insurance Buffer:

* Patient Registration
* Inpatient/Outpatient Data (Future Appointments, Event Capture functions)
* Insurance Data (Request Electronic Insurance Option, HMS data extract, VPS Kiosk, Remote Insurance Query, Insurance Verifier)

#2

The entries in the buffer are processed real time (sent to the payer asking about insurance eligibility status and existing benefits) under the following conditions:

1. Payer is locally and nationally active for eIV
2. All of the following fields must be populated:

* Insurance Company Name
* Patient Name
* Subscriber Identifier
* Patient Identifier (only necessary if the patient is not the subscriber)
* Insured/Subscriber’s Date of Birth (only necessary if the patient is not the subscriber)

#3

User researches buffer entries:

Depending on the buffer symbol of the entry that the user is reviewing, a user can quickly determine if the entry has been sent electronically to the payer asking for the patient’s eligibility status. The symbols are updated upon payers response to indicate active insurance or inactive insurance. The symbols also identify if there is a possible problem sending the entry or a possible issue with the payer’s answer; a user will need to correct or verify/research manually. After researching or reviewing the buffer entry, the user can either ‘accept’ or ‘reject’ the buffer entry to drop it off their worklist. Accepting an entry allows a user to update and verify a patient’s insurance record in the Patient file (#2).

#4

The eIV Nightly Process which runs at each VAMC has several procedures to perform:

1. The eIV buffer extract is run to pick up any entries in the buffer that are the direct result of the HMS file/extract and adds them to the IIV Transmission Queue file (#365.1) in the VistA database.
2. The eIV appointment extract is run to pick up any future appointments (10 days in the future) whose patient’s active insurance has not been verified within a predetermined amount of time called Freshness days (180 days). Any identified insurance that needs to be verified is added to the IIV Transmission Queue file (#365.1) in the VistA database.
3. Checks for any outstanding eIV inquiries that have not been answered by the payer yet, and determines if it is allowed to resend the inquiry or not. If not, it then determines if it is supposed to mark the outstanding inquiry as a communication failure or not.
4. Sends the eIV inquiries that were identified in steps a through c by looping through the outstanding entries in the IIV Transmission Queue file (#365.1). VistA transmits them electronically via HL7 through the Financial Service Center (FSC) located at the Austin Information Technology Center (AITC) and ultimately goes to the payer as an X12 270 transaction.

## Business Benefits

The changes to the Veterans Health Administration’s (VHA) VistA IB module will maintain compliance with legislative and federal mandates and will address gaps and inefficiencies in the current insurance verification process.

The implementation of the requested modifications will provide several benefits to the business users:

* Increase robustness of comments functionality and history for operations staff
* Enhanced security key functionality
* Providing improved and additional reporting capability to allow more effective management of the insurance data

## Assumptions and Constraints

### Design Assumptions

Design uses existing systems and infrastructure.

### Design Constraints

Design uses existing systems and infrastructure.

This effort has the following constraints:

* The solution will need to meet VA Enterprise Standards for development language, security, 508 compliance, application framework, and integration with other VA systems.
* The Health Insurance Portability and Accountability Act (HIPAA)

### Design Trade-offs

There are no design trade-offs with the chosen design.

## Overview of the Significant Requirements

### Overview of Significant Functional Requirements

|  |
| --- |
| System Feature: eIV Extract Logic  These requirements define the changes the eInsurance team wants done to the behavior of the existing eIV Insurance Buffer extract and the eIV appointment extract. |
| System Feature: Enhance VistA  These requirements define the changes the eInsurance team wants done to the behavior of the following areas:   * Upon patient’s date of death terminate any active insurance * Lock down user’s capabilities with editing insurance related data with existing keys * Expand the amount of data displayed when user views a patient’s benefit information * Modify what users can access from within the insurance buffer * Update the behavior of the eIV auto update process to not overwrite existing data * Update the display associated with patient policy information * Update the display of the addresses associated with the insurance companies * Update a few VistA functions that Insurance Capture Buffer calls, to add checks for the user’s keys * Update the existing trigger for sending messages from VistA to the National Insurance File (NIF) Interface * Add two new triggers for sending messages from VistA to the NIF Interface * Add an entry to the existing VistA files: TYPE OF COVERAGE (#355.2) and TYPE OF PLAN (#355.1) |
| System Feature: Insurance Comments  These requirements define the changes the eInsurance team wants done to the behavior of the patient policy comments with respect to the menu options: Patient Insurance Info View/Edit, Claims Tracking Edit options, and the Third Party Joint Inquiry (TPJI) option. It also covers changes that eInsurance team wants made with respect to the behavior of the group plan comments. |
| System Feature: IV Site Parameters  These requirements define the changes the eInsurance team wants done to the behavior of the Insurance Verification (IV) settings that can be accessed from VistA’s MCCR Site Parameters option. |
| System Feature: Reports  These requirements define the changes the eInsurance team wants done to the behavior of the reports listed below:   * Report – List Plans by Insurance Company * Report – Missing Data (\*New Report) * Report – Patients Without MEDICARE * Report – Active Policies with no Effective Date * Report – eIV Auto Update Report[[2]](#footnote-3) |
| System Feature: Worklist – Move Subscribers to a Different Plan  These requirements define the changes the eInsurance team wants done to the behavior of the existing VistA option: Move Subscribers to a Different Plan. |

Refer to the RSD for detailed functional requirements.

### Overview of Functional Workload / Performance Requirements

This effort does not include changes to the functional workload/performance requirements.

### Overview of Operational Requirements

This effort does not include changes to the operational requirements.

### Overview of the Technical Requirements

This effort does not include changes to the technical requirements.

### Overview of the Security or Privacy Requirements

This effort does not include changes to security or privacy of the existing VistA system.

### Overview of System Criticality and High Availability Requirements

There are no system criticality requirements or high availability requirements associated with this enhancement.

### Single Sign-on Requirement

Access is controlled by the VistA application and the underlying operating system. This effort does not mandate any additional access or sign-on requirements.

### Requirement for Use of Enterprise Portals

This modification does not include any use of enterprise portals.

### Special Device Requirements

This modification does not include any special device requirements.

## Legacy System Retirement

This modification does not retire any VistA legacy system.

# Conceptual Design

This project does not require any changes to the conceptual design, including the subsections below.

## Conceptual Application Design

### Application Context

This project does not require any changes to the application context.

### High-Level Application Design

This project does not require any changes to the high-level conceptual application design.

### Application Locations

This project does not require any changes to the application locations.

## Conceptual Data Design

### Project Conceptual Data Model

This project does not require any changes to the project conceptual data model.

### Database Information

Any changes to the database structure are documented in the DBMS Files (Section 5.1).

### User Interface Data Mapping

This project does not require any changes to the User Interface Data Mapping; therefore, all subsections do not apply to this effort.

#### Application Screen Interface

This project does not require any changes to the User Interface Data Mapping.

##### <Insert name of screen>

This project does not require any changes to the User Interface Data Mapping.

#### Application Report Interface

This project does not require any changes to the User Interface Data Mapping.

##### <Insert name of report>

This project does not require any changes to the User Interface Data Mapping.

#### Unmapped Data Element

This project does not require any changes to the User Interface Data Mapping.

## Conceptual Infrastructure Design

### System Criticality and High Availability

This is an existing VistA application. There are no changes to hardware configuration or transaction volume that would change current system availability.

### Special Technology

This project does not require any special technology.

### Technology Locations

This project does not require any changes to the technology locations.

### Conceptual Infrastructure Diagram

This project does not require any changes to the conceptual infrastructure diagram.

#### Location of Environments and External Interfaces

This project does not require any changes to the location of environments or location of the external interfaces.

#### Conceptual Production String Diagram

This project does not require any changes to the conceptual production string diagram.

# System Architecture

This project does not require any changes to the architecture.

## Hardware Architecture

This modification does not include any changes to the hardware architecture.

## Software Architecture

This modification does not include any changes to the software architecture.

## Network Architecture

This modification does not include any changes to the network architecture.

## Service Oriented Architecture / ESS

This modification does not include any changes to the service oriented architecture/ESS.

## Enterprise Architecture

This modification does not include any changes to the enterprise architecture.

# Data Design

## DBMS Files

This effort uses the standard FileMan database structures. Any changes to the database structures are documented in the Specific Requirements (Section 6.2.2). Below is a description of the database changes for this enhancement. Modifications to existing fields are bolded and highlighted to reflect where the modification occurs. New fields are reflected by the entire field definition being bolded and highlighted.

RSD 2.6.9.13

**2.342,.04 PERSON CONTACTED 0;4 FREE TEXT**

**INPUT TRANSFORM: K:$L(X)>30!($L(X)<3) X**

**LAST EDITED: APR 13, 2015**

**HELP-PROMPT: Answer must be 3-30 characters in length.**

**DESCRIPTION: This is the name of the person you contacted.**

**This is a free text name that can be entered.**

**It is recommended that you use the format of**

**Firstname MI Lastname, just as they would say**

**it to you.**

RSD 2.6.9.13

**2.342,.05 CONTACT PHONE # 0;5 FREE TEXT**

**INPUT TRANSFORM: K:$L(X)>20!($L(X)<7) X**

**LAST EDITED: APR 13, 2015**

**HELP-PROMPT: Answer must be 7-20 characters in length.**

**DESCRIPTION: This is the phone number of the person you**

**contacted.**

**If you contacted an insurance company and this**

**number is not in the insurance company file,**

**then you may want to enter it into that file as**

**well. The number entered here will only be**

**seen when looking at this contact.**

RSD 2.6.9.13

**2.342,.06 CALL REFERENCE NUMBER 0;6 FREE TEXT**

**INPUT TRANSFORM: K:$L(X)>35!($L(X)<3) X**

**LAST EDITED: APR 13, 2015**

**HELP-PROMPT: Answer must be 3-15 characters in length.**

**DESCRIPTION: If the company you called gave you a reference**

**number for the call then enter that reference**

**number here. Many companies will issue**

**reference numbers so that they can track their**

**calls and allow reference back to them by**

**others.**

RSD 2.6.9.13

**2.342,.07 METHOD OF CONTACT 0;7 SET**

**'1' FOR PHONE;**

**'2' FOR MAIL;**

**'3' FOR OVERNIGHT MAIL;**

**'4' FOR PERSONAL;**

**'5' FOR VOICE MAIL;**

**'6' FOR OTHER;**

**LAST EDITED: APR 13, 2015**

**DESCRIPTION: This is the method that you used to contact the**

**person contacted in this entry. Most contacts**

**will be by phone but many others will be by**

**mail.**

RSD 2.6.9.13

**2.342,.08 AUTHORIZATION NUMBER 0;8 FREE TEXT**

**INPUT TRANSFORM: K:$L(X)>35!($L(X)<3) X**

**LAST EDITED: APR 13, 2015**

**HELP-PROMPT: Answer must be 3-35 characters in length.**

**DESCRIPTION: Enter the treatment authorization number that**

**the insurance company gave you during this**

**contact.**

**The data in this field if it exists will be**

**considered the Treatment Authorization code for**

**this care and will automatically be used by the**

**billing module.**

RSD 2.6.5.2

2.312,3.04 POLICY NOT BILLABLE 3;4 SET

'0' FOR NO;

'1' FOR YES;

LAST EDITED: AUG 13, 1996

HELP-PROMPT: Enter NO if claims should not be created

against this policy.

DESCRIPTION: This field is used primarily for CHAMPUS

policies. If the patient is covered under

CHAMPUS, but it is known that claims should

never be submitted to the CHAMPUS Fiscal

Intermediary, then entering YES in this field

will cause Pharmacy claims to the FI not to be

created.

RSD 2.6.2.1

2,.351 DATE OF DEATH .35;1 DATE (audited)

INPUT TRANSFORM: S %DT="EPXT" D ^%DT S X=Y K:Y<1 X I $D(X) D H^D

GUTL K:X>DGTIME X K DGTIME,DGDATE I $D(X) S DFN

=DA D DEATH^DGLOCK,**DODUPT^IBCNUT(DFN,X)**

LAST EDITED: APR 14, 2011

HELP-PROMPT: Enter the date the patient died. Date must not

be before date of birth.

DESCRIPTION: Enter the date the patient died. Date needs to

be a precise date, i.e. a day, month, and year

MUST be included. Date of Death cannot be

prior to

P&T Effective Date

Date Ruled Incompetent (Civil and VA)

Date of Birth

AUDIT: YES, ALWAYS

NOTES: XXXX--CAN'T BE ALTERED EXCEPT BY PROGRAMMER

TRIGGERED by the DATE/TIME field of the PATIENT

MOVEMENT File

TRIGGERED by the MAS MOVEMENT TYPE field of the

PATIENT MOVEMENT File

RSD 2.6.7.1

36,3.01 TRANSMIT ELECTRONICALLY 3;1 SET

'0' FOR NO;

'1' FOR YES-LIVE;

'2' FOR YES-TEST;

INPUT TRANSFORM: **D SEND^IBCNHHLO(DA)** K:'$$EDIKEY^IBCNSC X

LAST EDITED: FEB 03, 2006

HELP-PROMPT: Enter 1 to bill electronically; 0 to print

bills for this insurance co; 2 to only transmit

for testing

DESCRIPTION: This is the flag that says whether or not an

insurance company is ready to be billed

electronically via 837/EDI functions.

NOTES: XXXX--CAN'T BE ALTERED EXCEPT BY PROGRAMMER

RSD 2.6.7.4 – Data Dictionary field

**350.9,TBD DAILY MSG TIME TBD;TBD FREE TEXT**

**INPUT TRANSFORM: K:(X'?4N)!('X)!(X>2400)!($E(X,3,4)'<60) X**

**HELP-PROMPT: This is the time of day to check to see if the**

**‘IB NIF TCP’ HL7 Logical Link is enabled. If problems are**

**encountered a message will be sent via MailMan to**

**mailgroup (TBD). The time must be in four digit**

**military format.**

**DESCRIPTION: Enter the time in four digit military format.**

**Examples: 0100 = 1 AM 1300 = 1 PM**

**NOTES: XXXX--CAN'T BE ALTERED EXCEPT BY PROGRAMMER**

RSD 2.6.7.5

36,.13 TYPE OF COVERAGE 0;13 POINTER TO TYPE OF INSURANCE COVERAGE

FILE (#355.2)

INPUT TRANSFORM: **D:X’=”” SEND^IBCNHHLO(DA)**

LAST EDITED: NOV 17, 2005

DESCRIPTION: If this insurance carrier provides only one type of coverage then

select the entry that best describes this carriers type of coverage. If this carrier

provides more than one type of coverage then select HEALTH INSURANCE.

The default answer if left unanswered is Health Insurance.

This is useful information when contacting carriers, when creating claims for

reimbursement, and when estimating if the payment received is appropriate.

If this field is answered it may affect choices that can be selected when entering

policy or benefit information.

FILES POINTED TO FIELDS

TYPE OF INSURANCE COVERAGE

(#355.2) TYPE OF COVERAGE (#.13)

RSD 2.6.4.7

**2.312,8.03 TRACE NUMBER 8;3 FREE TEXT**

**INPUT TRANSFORM: K:$L(X)>15!($L(X)<3) X**

**LAST EDITED: APR 08, 2015**

**HELP-PROMPT: Answer must be 3-15 characters in length.**

**DESCRIPTION: This field will contain the Trace Number**

**assigned by EC that is used for tracking a**

**message between EC and the vendor.**

**TECHNICAL DESCR: This field is updated from the Process Entry**

**(PE) action on the Insurance Buffer's Accept**

**Entry process. It is also updated by the**

**Insurance Buffer's Auto Update Process.**

**FILES POINTED TO FIELDS**

**X12 271 SERVICE TYPE (#365.013) REQUESTED SERVICE TYPE (#8.02)**

RSD 2.6.10.18

**350.9,51.27 270 MASTER SWITCH REALTIME 51;27 SET**

**'Y' FOR YES;**

**'N' FOR NO;**

**LAST EDITED: SEP 12, 2013**

**HELP-PROMPT: Enter ‘YES’ if real time 270 transactions can be created**

**and transmitted to the Eligibility Communicator (EC).**

**DESCRIPTION: A 'Y'es value indicates that real time 270 transactions can**

**be created and transmitted. A ‘N’o value indicates that realtime 270**

**transactions cannot be created and transmitted.**

RSD 2.6.10.22

**350.9,51.28 270 MASTER SWITCH NIGHTLY 51;28 SET**

**'Y' FOR YES;**

**'N' FOR NO;**

**LAST EDITED: SEP 12, 2013**

**HELP-PROMPT: Enter ‘YES’ if nightly extract 270 transactions can be**

**created and transmitted to the Eligibility Communicator (EC).**

**DESCRIPTION: A 'Y'es value indicates that nightly extracts 270 transactions**

**can be created and transmitted. A ‘N’o value indicates that nightly extracts**

**270 transactions cannot be created and transmitted.**

## Non-DBMS Files

This effort does not use Non-DBMS files.

## Data View

This effort uses the standard FileMan database structures. Any changes to the filing of data into the databases are documented in Specific Requirements (Section 6.2.2). See DBMS files (Section 5.1) for more information.

For a general overview of the files associated with this effort refer to the Entity Relationship Diagrams that are listed in References (Section 1.8).

# Detailed Design

## Hardware Detailed Design

This project does not require any changes to the existing hardware.

## Software Detailed Design

### Conceptual Design

#### Product Perspective

The IB module in VistA is a software package where third-party insurance claims are entered and managed. As part of this process, patient’s insurance information also needs to be entered into the IB module and periodically verified for accuracy. The modifications in this document are associated with the entering, managing, and verification of patient insurance. For the purpose of this document, the term “eInsurance” will be used to refer to the entering, managing, and verification of patient insurance in IB within VistA.

##### User Interfaces

The user interface with all eInsurance software is performed through existing and new VistA screens. VistA is a character-based application accessible through terminal emulator software resident on networked computers. VistA uses VA FileMan as its database manager and Caché/M as the programming language.

##### Hardware Interfaces

Communications between VistA sites and HIPAA EDI Services pass through the Local Area Networks (LANs) and across the Department of Veterans Affairs (VA) intranet. The Services communicate over the AITC LAN and through the private frame relay to the communications clearinghouse for this project. The clearinghouse communicates with the trading partner processors that are continually designated using previously negotiated contractual terms and conditions.

##### Software Interfaces

The following software must be installed prior to the release of this product.

* Kernel V. 8.0
* MailMan V. 8.0
* VA FileMan V. 22.0
* Health Level Seven (HL7) V. 1.6
* Integrated Billing V. 2.0
* Registration V.5.3
* InterSystems Caché Cube 2011.1.2
* IB\*2.0\*497
* IB\*2.0\*528

##### Communications Interfaces

The primary communication of verifying insurance in VistA is the submission of insurance eligibility inquiries (270 transactions) from the VistA sites to third-party payers via the Eligibility Communicator (EC) and a clearinghouse. The third-party payer’s eligibility response (271 transactions) follows the same path in reverse back to the VistA sites.

The NIF interface is used by the IB module in VistA to populate and maintain the proper Health Plan Identifier (HPID) and Other Entity Identifier (OEID) associated with the insurance companies that are defined in VistA. This interface is bi-directional between VistA and the NIF located in Austin.

##### Memory Constraints

There are no memory constraints associated with this project.

##### Special Operations

There are no special operations associated with this project.

#### Product Features

This design document covers the following features under Specific Requirements (Section 6.2.2):

* eIV Extract Logic
* Enhance VistA
* Insurance Comments
* IV Site Parameters
* Reports
* Worklist – Move Subscribers to a Different Plan

#### User Characteristics

In general, the resources that shall work with the IB software and eInsurance processes need to be knowledgeable in the areas of Health Care Eligibility Benefits, Integrated Billing, and Insurance payer/billing practices.

#### Dependencies and Constraints

The enhancements to VistA will be compliant with regulations mandated by HIPAA.

### Specific Requirements

#### Database Repository

Refer to DBMS Files (Section 5.1) for required database modifications.

#### System Feature: eIV Extract Logic

The following sections of this SDD are taken from the eInsurance Compliance Phase 3 RSD. Each individual requirement text as included in the RSD is included here for completeness and readability. The technical overview and technical design for each section will be included after the requirement text by the inclusion of one or more of the Design Element Tables.

RSD 2.6.1.1 eIV Extract Logic – The eIV batch extract shall run regardless if the eIV contact info is populated in the eIV Site Parameters.

RSD 2.6.1.2 eIV Extract Logic – The eIV appointment extract shall run regardless if the eIV contact info is populated in the eIV Site Parameters.

RSD 2.6.1.3 eIV Extract Logic – The eIV buffer and appointment extract shall be updated if needed so that they follow the following business rules:

1. If the Subscriber ID is present on the patient’s active policy, regardless if the Subscriber ID is required or not required by the payer, eIV shall create 1 verification eIV inquiry with the Subscriber ID on the inquiry.
2. If the Subscriber ID is \*NOT\* present on the patient’s active policy and the Subscriber ID is required by the payer, eIV shall either create a new insurance buffer entry or modify an existing insurance buffer entry to indicate that manual verification is required.
3. If the Subscriber ID is \*NOT\* present on the patient’s active policy and the Subscriber ID is \*NOT\* required by the payer, eIV shall create 1 verification eIV inquiry with no Subscriber ID populated on the inquiry.

RSD 2.6.1.4 eIV Extract Logic – The eIV appointment extract shall be updated to exclude processing active insurance companies whose TYPE OF COVERAGE (#36, .13) = MEDICAL; MEDICAID; TORT/FEASOR; WORKERS’ COMPENSATION; VA SPECIAL CLASS.

RSD 2.6.1.5 eIV Extract Logic – The eIV appointment extract shall be updated to exclude processing active insurance companies whose group’s TYPE OF PLAN (#355.3, .09) = MEDICAL; MEDICAID; TORT/FEASOR; WORKERS’ COMPENSATION; VA SPECIAL CLASS.

RSD 2.6.1.6 eIV Extract Logic – The eIV appointment extract shall use the appointment date as the date of service for deceased patients.

RSD 2.6.1.7 eIV Extract Logic – The eIV buffer extract shall use the date the entry was saved in the IIV Transmission Queue (#365.1) as the service date for deceased patients.

RSD 2.6.1.8 eIV Extract Logic – The "Request Electronic Insurance Inquiry" option shall allow inquiries of deceased patients.

>> No change required. Method PAT^IBCNEQU for patient selection already allows the selection of deceased patients.

RSD 2.6.1.9 eIV Extract Logic – The "Request Electronic Insurance Inquiry" option shall use the date of service entered by the user as the date of service on the inquiry (while honoring payer allowable ranges) for deceased patients.

>> No change required. Method NPINIT^IBCNEQU for patient selection already allows the selection of deceased patients.

RSD 2.6.1.10 eIV Extract Logic – The eIV appointment extract shall be updated to exclude processing active insurance companies whose TYPE OF COVERAGE (#36, .13) = MEDI-CAL; MEDICAID; TORT/FEASOR; WORKERS’ COMPENATION; VA SPECIAL CLASS. Therefore, the appointment extract will include all active insurance companies \* not \* listed here.

RSD 2.6.1.11 eIV Extract Logic – The eIV appointment extract shall be updated to exclude processing active insurance companies whose group’s TYPE OF PLAN (#355.3, .09) = AUTOMOBILE; MEDI-CAL; MEDICAID; TORT FEASOR; WORKER’S COMPENSATION INSURANCE; VA SPECIAL CLASS. Therefore, the appointment extract will include all active insurance companies \* not \* listed here.

#### Design Element Tables (eIV Extract Logic)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNEHLM** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.1.1, 2.6.1.2 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNEDE | | | | | HLNAME^HLFNC  INIT^HLFNC2  GENERATE^HLMA  GETSTAT^IBCNEDST  MGRP^IBCNEUT5  MSG^IBCNEUT5  TXT^IBCNEUT7 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNEHLM ;DAOU/ALA - HL7 Registration MFN Message ;10-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*184,251,300,416,438,497,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process the outgoing registration MFN message  ;  ; Variables  ; MCT = Lines of MailMan message counter  ; QFL = Quit flag  ; HL\* = HL7 package specific variables  ; TAXID = Tax ID  ; CNTCPH = Contact Phone  ; CNTCEM = Contact Email  ; FRSH = Freshness Days  ; MGRP = Mailgroup to email messages to  ; INACT = Inactive Insurance Flag  ; CNTC = Contact IEN  ; APP = Application  ; EVENT = HL7 Event  ; CODE = Values sent in the MFN message  ; IPP = IP Port  ; IPA = IP Address  ; RESP = Response Code  ; IHLP = Interface HL7 Processing Type  ; IHLT = Interface HL7 Batch Start Time  ; IHLS = Interface HL7 Batch Stop Time  ; IVER = Interface Version  ; TIMOUT = Timeout Days Site Parameter  ; RETRY = Retry Flag Site Parameter  ;  N IBPERSIST  S IBPERSIST="N" ; persistence flag - If "N", FSC will not use the statistics on the NTE segment  D REG  Q  ; EN1 ; TaskMan entry point  N IBPERSIST  S IBPERSIST="Y" ; persistence flag - If "Y", FSC will use NTE segment to update their copy of the site's stats  D REG  ; Purge the task record  S ZTREQ="@"  Q  ; REG ; Registration message for when a site installs  NEW TAXID,CNTCPH,CNTCEM,CNTCNM,IBCNE,FRSH,MGRP,INACT,IHLP,MFE,HLSAN  NEW IHLT,CNTC,APP,EVENT,CODE,EDT,MFN,HL,HLFS,HLECH,MCT,HLPROD,HLX,ID  NEW HLEID,IPP,IPA,IBCNEDAT,HLCS,HLINST,HLN,RESP,HLHDR,HLREP  NEW HLTYPE,HLQ,HLRESLT,IHLS,HLCDOM,HLCINS,HLCSTCP,HLIP,%I,ZMID  NEW VMFE,IVER,TIMOUT,RETRY         ; IB\*2.0\*506  K ^TMP("HLS",$J) S MCT=0,QFL=0  ;  ; Get data from IB Parameters File  S TAXID=$TR($P($G(^IBE(350.9,1,1)),U,5),"-",""),CNTCPH="",CNTCEM="",CNTCNM=""  S IBCNE=$G(^IBE(350.9,1,51))  S FRSH=$P(IBCNE,U,1),TIMOUT=$P(IBCNE,U,5),RETRY=$P(IBCNE,U,26) ; IB\*2.0\*506  S MGRP=$$MGRP^IBCNEUT5()  S INACT=$E($$GET1^DIQ(350.9,"1,",51.08,"E"))  S IHLP=$P(IBCNE,U,13),IHLT=$P(IBCNE,U,14),CNTC=$P(IBCNE,U,16)  S IHLS=$P(IBCNE,U,19)  S IVER="6"  ;  I IHLP="I" S (IHLT,IHLS)=""  ;  ; Get contact specific information  I CNTC'="" D  . S CNTCNM=$P($G(^VA(200,CNTC,0)),U,1)  . S CNTCPH=$P($G(^VA(200,CNTC,.13)),U,2)  . S CNTCEM=$P($G(^VA(200,CNTC,.15)),U,1)  ;  ; Email if any missing data  I CNTC="" S MCT=MCT+1,MSG(MCT)="The Contact Person is not defined in the eIV Site Parameters. ",QFL=1  I CNTC'="",CNTCPH="" S MCT=MCT+1,MSG(MCT)="The office phone number of the eIV Contact Person is not defined (File 200, Field .132). ",QFL=1  I CNTC'="",CNTCEM="" S MCT=MCT+1,MSG(MCT)="The email address of the eIV Contact Person is not defined (File 200, Field .151). ",QFL=1  ;  I IHLP="B",IHLT=""!(IHLS="") D  S QFL=1  . S MCT=MCT+1,MSG(MCT)="The ""HL7 Response Processing Method"" selected is Batch but the HL7 Batch "  . I IHLT="",IHLS="" S MSG(MCT)=MSG(MCT)\_"Start and End Times are blank. " Q  . S MSG(MCT)=MSG(MCT)\_$S(IHLT="":"Start",1:"End")\_" Time is blank. "  ;  I FRSH=""!(INACT="")!(IHLP="") D  . S MCT=MCT+1,MSG(MCT)="The following eIV Site Parameters are not defined: "  . I FRSH="" S MCT=MCT+1,MSG(MCT)="""Days between electronic re-verification checks"" is blank. "  . I INACT="" S MCT=MCT+1,MSG(MCT)="""Look at a patient's inactive insurance?"" is blank. "  . I IHLP="" S MCT=MCT+1,MSG(MCT)="""HL7 Response Processing Method"" is blank. "  . Q  ;  I $O(MSG(""))'="" D MLMN  I QFL=1 Q  ; HL ; When a site installs, the enrollment should be an  ; "MUP" (update) record.  N DSTAT,VNTE,VZRR  S MFE(1)="MUP"  ;  ; Initialize the HL7  D INIT^HLFNC2("IBCNE IIV REGISTER",.HL)  S HLFS=HL("FS"),HLECH=HL("ECH"),HL("SAF")=$P($$SITE^VASITE,U,2,3),HLREP=$E(HL("ECH"),2)  ; S HLEID=$$HLP^IBCNEHLU("IBCNE IIV REGISTER")  ;  ; Set the MFI segment  S ID="Facility Table",APP="",EVENT="UPD",RESP="NE"  S ^TMP("HLS",$J,1)=$$MFI^VAFHLMFI(ID,APP,EVENT,,,RESP)  ;  ; Set the MFE segment  S EVENT=MFE(1),MFN="",EDT=$$DT^XLFDT()  S CODE=$P($$SITE^VASITE,U,3)\_$E(HLECH)  S VMFE=$$MFE^VAFHLMFE(EVENT,MFN,EDT,CODE)  S ^TMP("HLS",$J,2)=VMFE\_HLFS\_"CE"  ;  ; Set the ZRR segment  S VZRR="ZRR"\_HLFS\_"1"\_HLFS\_TAXID\_HLFS\_HLFS\_$$HLNAME^HLFNC(CNTCNM,$E(HLECH))\_"^C"\_HLFS  S VZRR=VZRR\_CNTCPH\_$E(HLECH)\_$E(HLECH)\_$E(HLECH)\_CNTCEM\_HLFS\_FRSH\_HLFS\_IHLP\_HLFS\_IHLT\_$E(HLECH)\_IHLS\_HLFS\_INACT\_HLFS\_IVER  S ^TMP("HLS",$J,3)=VZRR  ;  ; Set the NTE segment  S DSTAT=$$GETSTAT^IBCNEDST()  S VNTE="NTE"\_HLFS\_"1"\_HLFS\_HLFS\_IBPERSIST\_HLREP\_$TR(DSTAT,U,HLREP)  S VNTE=VNTE\_HLREP\_RETRY\_HLREP\_TIMOUT          ;IB\*2.0\*506  S ^TMP("HLS",$J,4)=VNTE  ;  D GENERATE^HLMA("IBCNE IIV REGISTER","GM",1,.HLRESLT,"")  I $P(HLRESLT,U,2)]"" S HLRESLT="Error - "\_$P(HLRESLT,U,2,99) D  Q  . S MSG(1)="HL7 eIV Registration Message not created."  . S MSG(2)=HLRESLT  . D MLMN  K ^TMP("HLS",$J)  Q  ; MLMN ; MailMan Message  D TXT^IBCNEUT7("MSG")  S XMSUB="eIV Registration Failure"  D MSG^IBCNEUT5(MGRP,XMSUB,"MSG(")  K XMSUB,XMY,MSG,XMZ,XMDUZ  Q. |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNEHLM ;DAOU/ALA - HL7 Registration MFN Message ;10-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*184,251,300,416,438,497,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process the outgoing registration MFN message  ;  ; Variables  ; MCT = Lines of MailMan message counter  ; QFL = Quit flag  ; HL\* = HL7 package specific variables  ; TAXID = Tax ID  ; CNTCPH = Contact Phone **; CNTCEM = Contact Email –**  **; CNTCNM = Contact Name**  ; FRSH = Freshness Days  ; MGRP = Mailgroup to email messages to  ; INACT = Inactive Insurance Flag **; CNTC = Contact IEN – 2.6.1.1, 2.6.1.2, Remove this line**  ; APP = Application  ; EVENT = HL7 Event  ; CODE = Values sent in the MFN message  ; IPP = IP Port  ; IPA = IP Address  ; RESP = Response Code  ; IHLP = Interface HL7 Processing Type  ; IHLT = Interface HL7 Batch Start Time  ; IHLS = Interface HL7 Batch Stop Time  ; IVER = Interface Version  ; TIMOUT = Timeout Days Site Parameter  ; RETRY = Retry Flag Site Parameter  ;  N IBPERSIST  S IBPERSIST="N" ; persistence flag - If "N", FSC will not use the statistics on the NTE segment  D REG  Q  ; EN1 ; TaskMan entry point  N IBPERSIST  S IBPERSIST="Y" ; persistence flag - If "Y", FSC will use NTE segment to update their copy of the site's stats  D REG  ; Purge the task record  S ZTREQ="@"  Q  ; REG ; Registration message for when a site installs  NEW TAXID,CNTCPH,CNTCEM,CNTCNM,IBCNE,FRSH,MGRP,INACT,IHLP,MFE,HLSAN **; 2.6.1.1, 2.6.1.2** NEW IHLT,**~~CNTC~~**,APP,EVENT,CODE,EDT,MFN,HL,HLFS,HLECH,MCT,HLPROD,HLX,ID  NEW HLEID,IPP,IPA,IBCNEDAT,HLCS,HLINST,HLN,RESP,HLHDR,HLREP  NEW HLTYPE,HLQ,HLRESLT,IHLS,HLCDOM,HLCINS,HLCSTCP,HLIP,%I,ZMID  NEW VMFE,IVER,TIMOUT,RETRY         ; IB\*2.0\*506  K ^TMP("HLS",$J) S MCT=0,QFL=0  ;  ; Get data from IB Parameters File  S TAXID=$TR($P($G(^IBE(350.9,1,1)),U,5),"-",""),CNTCPH="",CNTCEM="",CNTCNM=""  S IBCNE=$G(^IBE(350.9,1,51))  S FRSH=$P(IBCNE,U,1),TIMOUT=$P(IBCNE,U,5),RETRY=$P(IBCNE,U,26) ; IB\*2.0\*506  S MGRP=$$MGRP^IBCNEUT5()  S INACT=$E($$GET1^DIQ(350.9,"1,",51.08,"E"))  S IHLP=$P(IBCNE,U,13),IHLT=$P(IBCNE,U,14),CNTC=$P(IBCNE,U,16)  S IHLS=$P(IBCNE,U,19)  S IVER="6"  ;  I IHLP="I" S (IHLT,IHLS)=""  ; **~~; Get contact specific information~~ 2.6.1.1, 2.6.1.2 ~~I CNTC'="" D  . S CNTCNM=$P($G(^VA(200,CNTC,0)),U,1)  . S CNTCPH=$P($G(^VA(200,CNTC,.13)),U,2)  . S CNTCEM=$P($G(^VA(200,CNTC,.15)),U,1~~)**  ;  ; Email if any missing data **;2.6. 1.1, 2.6. 1.2** **~~I CNTC="" S MCT=MCT+1,MSG(MCT)="The Contact Person is not defined in the eIV Site Parameters. ",QFL=1  I CNTC'="",CNTCPH="" S MCT=MCT+1,MSG(MCT)="The office phone number of the eIV Contact Person is not defined (File 200, Field .132). ",QFL=1  I CNTC'="",CNTCEM="" S MCT=MCT+1,MSG(MCT)="The email address of the eIV Contact Person is not defined (File 200, Field .151). ",QFL=1~~**  ;  I IHLP="B",IHLT=""!(IHLS="") D  S QFL=1  . S MCT=MCT+1,MSG(MCT)="The ""HL7 Response Processing Method"" selected is Batch but the HL7 Batch "  . I IHLT="",IHLS="" S MSG(MCT)=MSG(MCT)\_"Start and End Times are blank. " Q  . S MSG(MCT)=MSG(MCT)\_$S(IHLT="":"Start",1:"End")\_" Time is blank. "  ;  I FRSH=""!(INACT="")!(IHLP="") D  . S MCT=MCT+1,MSG(MCT)="The following eIV Site Parameters are not defined: "  . I FRSH="" S MCT=MCT+1,MSG(MCT)="""Days between electronic re-verification checks"" is blank. "  . I INACT="" S MCT=MCT+1,MSG(MCT)="""Look at a patient's inactive insurance?"" is blank. "  . I IHLP="" S MCT=MCT+1,MSG(MCT)="""HL7 Response Processing Method"" is blank. "  . Q  ;  I $O(MSG(""))'="" D MLMN  I QFL=1 Q  ; HL ; When a site installs, the enrollment should be an  ; "MUP" (update) record.  N DSTAT,VNTE,VZRR  S MFE(1)="MUP"  ;  ; Initialize the HL7  D INIT^HLFNC2("IBCNE IIV REGISTER",.HL)  S HLFS=HL("FS"),HLECH=HL("ECH"),HL("SAF")=$P($$SITE^VASITE,U,2,3),HLREP=$E(HL("ECH"),2)  ; S HLEID=$$HLP^IBCNEHLU("IBCNE IIV REGISTER")  ;  ; Set the MFI segment  S ID="Facility Table",APP="",EVENT="UPD",RESP="NE"  S ^TMP("HLS",$J,1)=$$MFI^VAFHLMFI(ID,APP,EVENT,,,RESP)  ;  ; Set the MFE segment  S EVENT=MFE(1),MFN="",EDT=$$DT^XLFDT()  S CODE=$P($$SITE^VASITE,U,3)\_$E(HLECH)  S VMFE=$$MFE^VAFHLMFE(EVENT,MFN,EDT,CODE)  S ^TMP("HLS",$J,2)=VMFE\_HLFS\_"CE"  ;  ; Set the ZRR segment  **; 2.6.1.1, 2.6.1.2, Add following line**  **S (CNTCPH,CNTCEM,CNTCNM)=”” ; don’t send contact info & don’t change msg layout ;IB\*549**  S VZRR="ZRR"\_HLFS\_"1"\_HLFS\_TAXID\_HLFS\_HLFS\_$$HLNAME^HLFNC(CNTCNM,$E(HLECH))\_"^C"\_HLFS  S VZRR=VZRR\_CNTCPH\_$E(HLECH)\_$E(HLECH)\_$E(HLECH)\_CNTCEM\_HLFS\_FRSH\_HLFS\_IHLP\_HLFS\_IHLT\_$E(HLECH)\_IHLS\_HLFS\_INACT\_HLFS\_IVER  S ^TMP("HLS",$J,3)=VZRR  ;  ; Set the NTE segment  S DSTAT=$$GETSTAT^IBCNEDST()  S VNTE="NTE"\_HLFS\_"1"\_HLFS\_HLFS\_IBPERSIST\_HLREP\_$TR(DSTAT,U,HLREP)  S VNTE=VNTE\_HLREP\_RETRY\_HLREP\_TIMOUT          ;IB\*2.0\*506  S ^TMP("HLS",$J,4)=VNTE  ;  D GENERATE^HLMA("IBCNE IIV REGISTER","GM",1,.HLRESLT,"")  I $P(HLRESLT,U,2)]"" S HLRESLT="Error - "\_$P(HLRESLT,U,2,99) D  Q  . S MSG(1)="HL7 eIV Registration Message not created."  . S MSG(2)=HLRESLT  . D MLMN  K ^TMP("HLS",$J)  Q  ; MLMN ; MailMan Message  D TXT^IBCNEUT7("MSG")  S XMSUB="eIV Registration Failure"  D MSG^IBCNEUT5(MGRP,XMSUB,"MSG(")  K XMSUB,XMY,MSG,XMZ,XMDUZ  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNEDE2** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.1.3, 2.6.1.4, 2.6.1.5, 2.6.1.6, 2.6.1.10, 2.6.1.11 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNEDE  IBCNEHL1 | | | | | SETTINGS^IBCNEDE7 SDAPI^SDMA301  ALL^IBCNS1  EXCLUDE^IBCNEUT4  INSERROR^IBCNEUT3  PYRACTV^IBCNEDE7  BFEXIST^IBCNEUT5  PT^IBCNEBF  UPDT^IBCNEDE6  TQUPDSV^IBCNEUT5  ADDTQ^IBCNEUT5  SIDCHK^IBCNEDE5  SETTQ^IBCNEDE7  MGRP^IBCNEUT5  MSG^IBCNEUT5 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  EN ; Loop through designated cross-references for updates  ; Pre reg extract (Appointment extract)  ;  .  .   .. ;  .. ; Loop through dates in range at clinic  .. F  S APTDT=$O(^TMP($J,"SDAMA301",CLNC,DFN,APTDT)) Q:('APTDT)!((APTDT\1)>ENDDT)!(CNT'<MAXCNT) D  Q:$G(ZTSTOP)  ... ;  ... S SRVICEDT=APTDT\1 ;Set service date equal to appointment date  ... S FRESHDT=$$FMADD^XLFDT(SRVICEDT,-FRESHDAY)  ... ;  ... ; Update count for periodic check  ... S IBCNETOT=IBCNETOT+1  ... ; Check for request to stop background job, periodically  ... I $D(ZTQUEUED),IBCNETOT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  ... ;  ... S IBSDATA=$G(^TMP($J,"SDAMA301",CLNC,DFN,APTDT))  ... S ELG=$P(IBSDATA,U,8)  ... S ELG=$S(ELG'="":ELG,1:$P($G(^DPT(DFN,.36)),U,1))  ... I $P($G(^DPT(DFN,0)),U,21) Q         ; Exclude if test patient  ... I $P($G(^DPT(DFN,.35)),"^",1)'="" Q  ; Exclude if patient is deceased  ... ;  ... D ELG Q:'OK     ; Check for eligibility exclusion  ... ;  ... K ACTINS  ... D ALL^IBCNS1(DFN,"ACTINS",2)  ... ;  ... I '$D(ACTINS(0)) Q  ; Patient has no active ins  ... ;  ... S INREC=0 ; Record ien  ... F  S INREC=$O(ACTINS(INREC)) Q:('INREC)!(CNT'<MAXCNT) D  ... . S INSIEN=$P($G(ACTINS(INREC,0)),U,1) ; Insurance ien  ... . S INSNAME=$P($G(^DIC(36,INSIEN,0)),U)  ... . ; exclude policies that have been verified within "freshness days"  ... . S VDATE=$P($G(ACTINS(INREC,1)),U,3)  ... . I VDATE'="",SRVICEDT'>$$FMADD^XLFDT(VDATE,FRESHDAY) Q  ... . ; allow only one MEDICARE transmission per patient  ... . I INSNAME["MEDICARE",MCAREFLG Q  ... . ; exclude pharmacy policies  ... . I $$GET1^DIQ(36,INSIEN\_",",.13)="PRESCRIPTION ONLY" Q  ... . S GIEN=+$P($G(ACTINS(INREC,0)),U,18)  ... . I GIEN,$$GET1^DIQ(355.3,GIEN\_",",.09)="PRESCRIPTION" Q  ... . ; check for ins. to exclude (i.e. Medicaid)  ... . I $$EXCLUDE^IBCNEUT4(INSNAME) Q  ... . ; check insurance policy expiration date  ... . I $$EXPIRED($P($G(ACTINS(INREC,0)),U,4)) Q  ... . ;  ... . ; set patient id field IB\*2\*416  ... . S PATID=$P($G(ACTINS(INREC,5)),U,1) ; 5.01 field  ... . ;  ... . S PAYERSTR=$$INSERROR^IBCNEUT3("I",INSIEN) ; Get payer info.  .  .  .   ... . I SIDACT=4 D SET("","",PATID) S:INSNAME["MEDICARE" MCAREFLG=1   ... . Q  ... Q ENQ K ^TMP($J,"SDAMA301"),^TMP("IBCNEDE2",$J)  Q  ; CLINICEX ; Clinic exclusion  .  . |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  EN ; Loop through designated cross-references for updates  ; Pre reg extract (Appointment extract)  ;  .  .   .. ;  .. ; Loop through dates in range at clinic  .. F  S APTDT=$O(^TMP($J,"SDAMA301",CLNC,DFN,APTDT)) Q:('APTDT)!((APTDT\1)>ENDDT)!(CNT'<MAXCNT) D  Q:$G(ZTSTOP)  ... ;  ... S SRVICEDT=APTDT\1 ;Set service date equal to appointment date  ... S FRESHDT=$$FMADD^XLFDT(SRVICEDT,-FRESHDAY)  ... ;  ... ; Update count for periodic check  ... S IBCNETOT=IBCNETOT+1  ... ; Check for request to stop background job, periodically  ... I $D(ZTQUEUED),IBCNETOT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  ... ;  ... S IBSDATA=$G(^TMP($J,"SDAMA301",CLNC,DFN,APTDT))  ... S ELG=$P(IBSDATA,U,8)  ... S ELG=$S(ELG'="":ELG,1:$P($G(^DPT(DFN,.36)),U,1))  ... I $P($G(^DPT(DFN,0)),U,21) Q         ; Exclude if test patient **Remove the following line 2.6.1.6**  **~~... I $P($G(^DPT(DFN,.35)),"^",1)'="" Q  ; Exclude if patient is deceased~~**  ... ;  ... D ELG Q:'OK     ; Check for eligibility exclusion  ... ;  ... K ACTINS  ... D ALL^IBCNS1(DFN,"ACTINS",2)  ... ;  ... I '$D(ACTINS(0)) Q  ; Patient has no active ins  ... ;  ... S INREC=0 ; Record ien  ... F  S INREC=$O(ACTINS(INREC)) Q:('INREC)!(CNT'<MAXCNT) D  ... . S INSIEN=$P($G(ACTINS(INREC,0)),U,1) ; Insurance ien  ... . S INSNAME=$P($G(^DIC(36,INSIEN,0)),U) **... .S ZZ=$$GET1^DIQ(36,INSIEN\_",",.13,"I") ; 2.6.1.10 Type of Coverage  ... .I ZZ'=14,ZZ'=10,ZZ'=19,ZZ'=13,ZZ'=20 Q         ; 2.6.1.10 Exclude all other types of coverage**   ... . ; exclude policies that have been verified within "freshness days"  ... . S VDATE=$P($G(ACTINS(INREC,1)),U,3)  ... . I VDATE'="",SRVICEDT'>$$FMADD^XLFDT(VDATE,FRESHDAY) Q  ... . ; allow only one MEDICARE transmission per patient  ... . I INSNAME["MEDICARE",MCAREFLG Q  **Remove the following lines 2.6.1.4**  **~~... . ; exclude pharmacy policies  ... . I $$GET1^DIQ(36,INSIEN\_",",.13)="PRESCRIPTION ONLY" Q~~**  ... . S GIEN=+$P($G(ACTINS(INREC,0)),U,18)**... .S ZZ=$$GET1^DIQ(355.32,GIEN\_",",.09,"I") ; 2.6.1.11 Type of Plan  ... .I ZZ'=39,ZZ'=17,ZZ'=30,ZZ'=16,ZZ'=29, ZZ'=66, ZZ'=54, Q         ; 2.6.1.11 Exclude all other types of plan**  **Remove the following line 2.6.1.5**  **~~... . I GIEN,$$GET1^DIQ(355.3,GIEN\_",",.09)="PRESCRIPTION" Q~~**  ... . ; check for ins. to exclude (i.e. Medicaid)  ... . I $$EXCLUDE^IBCNEUT4(INSNAME) Q  ... . ; check insurance policy expiration date  ... . I $$EXPIRED($P($G(ACTINS(INREC,0)),U,4)) Q  ... . ;  ... . ; set patient id field IB\*2\*416  ... . S PATID=$P($G(ACTINS(INREC,5)),U,1) ; 5.01 field  ... . ;  ... . S PAYERSTR=$$INSERROR^IBCNEUT3("I",INSIEN) ; Get payer info  .  .  **... . ; 2.6.1.3 - INREQTQ set in IBCNEDE5**  ... . I SIDACT=4 D SET("",**INRECTQ**,PATID) S:INSNAME["MEDICARE" MCAREFLG=1   ... . Q  ... Q ENQ K ^TMP($J,"SDAMA301"),^TMP("IBCNEDE2",$J)  Q  ; CLINICEX ; Clinic exclusion  .  . |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNEDE1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.1.7 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNEDE  IBCNERTQ | | | | | SETTINGS^IBCNEDE7  SYMBOL^IBCNBLL  INSERROR^IBCNEUT3  PYRACTV^IBCNEDE7  BUFF^IBCNEUT2  CLEAR^IBCNEUT4  UPDDTS^IBCNEDE6  TQUPDSV^IBCNEUT5  SETSTC^IBCNERTQ  SIDCHK^IBCNEDE5  SETTQ^IBCNEDE7 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .   ;  S LOOPDT="" ; Date used to loop through the IB global  F  S LOOPDT=$O(^IBA(355.33,"AEST","E",LOOPDT)) Q:LOOPDT=""!(CNT=MAXCNT) D  Q:$G(ZTSTOP)  . S IEN=""  . F  S IEN=$O(^IBA(355.33,"AEST","E",LOOPDT,IEN)) Q:IEN=""!(CNT=MAXCNT) D  Q:$G(ZTSTOP)  .. ; Update count for periodic check  .. S IBCNETOT=IBCNETOT+1  .. ; Check for request to stop background job, periodically  .. I $D(ZTQUEUED),IBCNETOT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  .. ;  .. ; Get symbol, if symbol'=" " OR "!" then quit  .. S ISYMBOL=$$SYMBOL^IBCNBLL(IEN) ; Insurance buffer symbol  .. I (ISYMBOL'=" ")&(ISYMBOL'="!") Q  .. ;  .. ; Don't extract ePharmacy buffer entries - IB\*2\*435  .. I +$P($G(^IBA(355.33,IEN,0)),U,17) Q  .. ;  .. ; Get the eIV STATUS IEN and quit for response related errors  .. S STATIEN=+$P($G(^IBA(355.33,IEN,0)),U,12)  .. I ",11,12,15,"[(","\_STATIEN\_",") Q  ; Prevent update for response errors  .. ;  .. S OVRFRESH=$P($G(^IBA(355.33,IEN,0)),U,13) ; Freshness OvrRd flag  .. S DFN=$P($G(^IBA(355.33,IEN,60)),U,1) ; Patient DFN  .. Q:DFN=""  .. I $P($G(^DPT(DFN,0)),U,21) Q           ; Exclude if test patient  .. ;  .. S PDOD=$P($G(^DPT(DFN,.35)),U,1)\1 ; Patient's date of death  .. S SRVICEDT=+$P($G(^IBA(355.33,IEN,0)),U,18) S:'SRVICEDT SRVICEDT=DT ; Service Date   .. I PDOD,PDOD<SRVICEDT S SRVICEDT=PDOD  .. S FRESHDT=$$FMADD^XLFDT(SRVICEDT,-FRESHDAY)  .. S PAYERSTR=$$INSERROR^IBCNEUT3("B",IEN) ; Payer String  .. S PAYERID=$P(PAYERSTR,U,3),PIEN=$P(PAYERSTR,U,2) ; Payer ID  .. S SYMBOL=+PAYERSTR                               ; Payer Symbol  .. I '$$PYRACTV^IBCNEDE7(PIEN) Q          ; Payer is not nationally active  .. ;.  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  .   ;  S LOOPDT="" ; Date used to loop through the IB global  F  S LOOPDT=$O(^IBA(355.33,"AEST","E",LOOPDT)) Q:LOOPDT=""!(CNT=MAXCNT) D  Q:$G(ZTSTOP)  . S IEN=""  . F  S IEN=$O(^IBA(355.33,"AEST","E",LOOPDT,IEN)) Q:IEN=""!(CNT=MAXCNT) D  Q:$G(ZTSTOP)  .. ; Update count for periodic check  .. S IBCNETOT=IBCNETOT+1  .. ; Check for request to stop background job, periodically  .. I $D(ZTQUEUED),IBCNETOT#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  .. ;  .. ; Get symbol, if symbol'=" " OR "!" then quit  .. S ISYMBOL=$$SYMBOL^IBCNBLL(IEN) ; Insurance buffer symbol  .. I (ISYMBOL'=" ")&(ISYMBOL'="!") Q  .. ;  .. ; Don't extract ePharmacy buffer entries - IB\*2\*435  .. I +$P($G(^IBA(355.33,IEN,0)),U,17) Q  .. ;  .. ; Get the eIV STATUS IEN and quit for response related errors  .. S STATIEN=+$P($G(^IBA(355.33,IEN,0)),U,12)  .. I ",11,12,15,"[(","\_STATIEN\_",") Q  ; Prevent update for response errors  .. ;  .. S OVRFRESH=$P($G(^IBA(355.33,IEN,0)),U,13) ; Freshness OvrRd flag  .. S DFN=$P($G(^IBA(355.33,IEN,60)),U,1) ; Patient DFN  .. Q:DFN=""  .. I $P($G(^DPT(DFN,0)),U,21) Q           ; Exclude if test patient  .. ;  .. S PDOD=$P($G(^DPT(DFN,.35)),U,1)\1 ; Patient's date of death  .. S SRVICEDT=+$P($G(^IBA(355.33,IEN,0)),U,18) S:'SRVICEDT SRVICEDT=DT ; Service Date  **..; 2.6.1.7 Removing the next line**  **.. ;I PDOD,PDOD<SRVICEDT S SRVICEDT=PDOD** .. S FRESHDT=$$FMADD^XLFDT(SRVICEDT,-FRESHDAY)  .. S PAYERSTR=$$INSERROR^IBCNEUT3("B",IEN) ; Payer String  .. S PAYERID=$P(PAYERSTR,U,3),PIEN=$P(PAYERSTR,U,2) ; Payer ID  .. S SYMBOL=+PAYERSTR                               ; Payer Symbol  .. I '$$PYRACTV^IBCNEDE7(PIEN) Q          ; Payer is not nationally active  .. ;.  .  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNEDE5 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.1.3 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNEDE1  IBCNEDE2  IBCNEDE3  IBCNEDE4  IBCNEDE6  IBCNEDEQ | | | | | INSERROR^IBCNEUT3 PYRAPP^IBCNEUT5 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  SIDC1 ;  S SIDACT=$S(SIDREQ:3,1:4)   ; SIDCHKX ; EXIT POINT  ;  Q SIDACT\_U\_SIDCNT  ;.  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  SIDC1 ;  S SIDACT=$S(SIDREQ:3,1:4)  **S:SIDACT=4 INREQTQ=[CODE]  ;2.6.1.3 Set to INREC to be built in TQ**  ; SIDCHKX ; EXIT POINT  ;  Q SIDACT\_U\_SIDCNT  ;. | | | | | | | | | |

#### System Feature: Enhance VistA

RSD 2.6.2.1 Enhance VistA (DOD) - When a patient is flagged as deceased, VistA shall automatically set the expiration date on all active policies (if the effective date is populated) associated with that patient as the date of death +1 day.

RSD 2.6.2.2 Enhance VistA (DOD) – As a one-time occurrence, upon installation of the VistA patch, all active policies (if the effective date is populated) associated with deceased patients shall be modified to set the expiration date equal to the patient's date of death +1 day.

RSD 2.6.2.3 Enhance VistA (DOD) - The Patient Policy Information Screen shall display the patient's date of death (if populated) in the screen header.

RSD 2.6.2.4 Enhance VistA (DOD) - As a one-time occurrence, upon installation of the VistA patch, a mailman message shall be sent identifying the patient's and policies that were automatically termed as a result of installing the patch.

RSD 2.6.3.1 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit COVERAGE LIMITATIONS (action CV). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by Coverage Limitations action (CV).

RSD 2.6.3.2 Enhance VistA (Key) - Within the Patient Insurance Info View/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit COVERAGE LIMITATIONS (action CV). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by Coverage Limitations action (CV).

RSD 2.6.3.3 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit ANNUAL BENEFITS (action AB). There are two places (paths) that need to be updated. Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by Annual Benefits action (AB). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by Annual Benefits action (AB).

RSD 2.6.3.4 Enhance VistA (Key) - Within the Patient Insurance Info View/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit ANNUAL BENEFITS (action AB). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by Annual Benefits action (AB).

RSD 2.6.3.5 Enhance VistA (Key) - Within the Patient Insurance Info View/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may CHANGE PLAN INFO (action PI). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by Change Plan Info action (PI).

RSD 2.6.3.6 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may CHANGE PLAN INFO (action PI). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by Change Plan Info action (PI).

RSD 2.6.3.7 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit UR INFO (action UI). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by UR Info action (UI).

RSD 2.6.3.8 Enhance VistA (Key) - Within the Patient Insurance Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may edit UR INFO (action UI). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by UR Info action (UI).

RSD 2.6.3.9 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may INACTIVATE PLAN (action IP). There are two places (paths) that need to be updated. Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by Inactivate Plan action (IP). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by Inactivate Plan action (IP).

RSD 2.6.3.10 Enhance VistA (Key) - Within the Patient Insurance option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may INACTIVATE PLAN (action IP). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by Inactivate Plan action (IP).

RSD 2.6.3.11 Enhance VistA (Key) - Within the Insurance Company Entry/Edit option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may EDIT COMMENTS (action PC). Path: Insurance Company Entry/Edit option (EI), followed by View Plans (VP), followed by View/Edit Plan action (VP), followed by Edit Comments action (PC).

RSD 2.6.3.12 Enhance VistA (Key) - Within the Patient Insurance option, VistA shall be restricted so that only users with the "IB GROUP PLAN EDIT" security key may FAST EDIT ALL plan specific information (action EA). Path: Patient Insurance Info View/Edit option (PI), followed by View Policy action (VP), followed by Fast Edit All action (EA).

RSD 2.6.4.1 Enhance VistA (eIV) - The Insurance Buffer's Expand Benefit action shall be modified to display the data elements that are found on the eIV Response Report that are currently not on the Expand Benefit action. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Expand Benefit action (EB).

RSD 2.6.4.2 Enhance VistA (eIV) – The Insurance Buffer’s eIV Response Report action shall be removed from the buffer processing screens. There are several paths that need to be updated. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Expand Entry action (EE), remove RR action. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Accept Entry action (AE), followed by Expand Entry action (EE), remove RR action. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Process Entry action (PE), followed by Expand Entry action (EE), remove RR action.

RSD 2.6.4.3 Enhance VistA (eIV) - The Insurance Buffer's Accept Entry process shall be modified to display the data elements that are found on the eIV Response Report that are currently not on the Expand Benefit action. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Accept Entry action (AE), followed by Expand Benefits action (BE).

RSD 2.6.4.4 Enhance VistA (eIV) - The Insurance Buffer's Accept Entry process shall be modified to allow a user to save to the Patient file (#2) the data elements that are found on the eIV Response Report that are currently not displayed in the Expand Benefit action. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Accept Entry action (AE), when user answers “yes” to prompt – “Save Benefits?”

RSD 2.6.4.5 Enhance VistA (eIV) - The Expand Benefits action (EB) within the Patient Insurance Info View\Edit option shall be modified to display the additional fields that the eIV Response Report has. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Expand Benefits action (EB).

RSD 2.6.4.6 Enhance VistA (eIV) - The Expand Benefits action (EB) within the Claims Tracking option(s) shall be modified to display the additional fields that the eIV Response Report has. Path: Claims Tracking Edit options (CT), followed by View Policy action (VP), followed by Expand Benefits action (EB).

RSD 2.6.4.7 Enhance VistA (eIV) - The Insurance Buffer shall display all Medicare entries to users regardless of the user's security keys and regardless of whether the patient has active policies or not. There are two places (paths) that need to be updated. Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), on the default main screen – Complete Buffer (CB). Path: Patient Insurance Menu (PI), followed by Process Insurance Buffer (BI), followed by Medicare Buffer (MB).

RSD 2.6.4.8 Enhance VistA (eIV) - When Medicare eIV responses are processed via eIV's auto update, only active policies (Part A and/or Part B) shall be updated.

RSD 2.6.4.9 Enhance VistA (eIV) - When eIV responses are processed via eIV's auto update, the following fields shall NOT be overwritten: Subscriber DOB (policy holder), Subscriber ID, Patient ID/Dependent ID, Subscriber Name/Name of Insurer, (Insured's Address 1, Address 2, City, State, Zip, Country, Country Subdivision), Coordination of Benefits, Subscriber SSN, Name of Insured, Pt. Relationship to Insured. \*If the data field is populated in #2.312; don’t change it and don’t overwrite it with null. If the data field is null in #2.312; save new data. “Fields” are to be treated separately with the exception of the Address fields. Treat address as a unit for the above rules. Address Unit = Insured's Address 1, Address 2, City, State, Zip, Country, and Country Subdivision.

RSD 2.6.4.10 Enhance VistA (eIV) - The HL7 Logical Link "IIV EC" shall be updated to use the domain name of the receiving site rather than the IP address.

RSD 2.6.5.1 Enhance VistA (IB) - Within the Patient Insurance Info View\Edit option, when the user selects the action 'Effective Dates' (ED) the user shall be prompted with "Is This Policy Billable?" rather than "Policy Not Billable".

RSD 2.6.5.2 Enhance VistA (IB) - VistA shall be updated so that the modification to the prompt from "Policy Not Billable" to "Is This Policy Billable", is reflected properly in the storage of the YES/NO answer. A user entered "No" shall be saved as "Yes" in the database and vice versa to keep the existing EDI functionality in the background working properly. \*Explain as comments in the code and the VistA database of why we are storing opposite value of what the user entered. (Original field as displayed to the user was a double negative thus causing confusion to the users.)

RSD 2.6.5.3 Enhance VistA (IB) - Within the Patient Insurance Info View\Edit option, in the 'Effective Dates' (ED) section the display shall show "Is This Policy Billable?" with the appropriate value (opposite of what is stored in the database) rather than "Policy Not Billable". \*Explain as comments in the code and the VistA database of why we are displaying opposite value of what is stored.

RSD 2.6.5.4 Enhance VistA (IB) - The View Insurance Company option shall display the address associated with the prescription under the Prescription Claims Office Information section if any part of the prescription address is populated.

RSD 2.6.5.5 Enhance VistA (IB) - The Insurance Company Entry/Edit option shall display the address associated with the prescription under the Prescription Claims Office Information section if any part of the prescription address is populated.

RSD 2.6.5.6 Enhance VistA (IB) - The View Insurance Company option shall display the address associated with the inpatient patient under the Inpatient Claims Office Information section if any part of the inpatient address is populated.

RSD 2.6.5.7 Enhance VistA (IB) - The Insurance Company Entry/Edit option shall display the address associated with the inpatient patient under the Inpatient Claims Office Information section if any part of the inpatient address is populated.

RSD 2.6.5.8 Enhance VistA (IB) - The View Insurance Company option shall display the address associated with the outpatient patient under the Outpatient Claims Office Information section if any part of the outpatient address is populated.

RSD 2.6.5.9 Enhance VistA (IB) - The Insurance Company Entry/Edit option shall display the address associated with the outpatient patient under the Outpatient Claims Office Information section if any part of the outpatient address is populated.

RSD 2.6.5.10 Enhance VistA (IB) - The View Insurance Company option shall display the address associated with appeals under the Appeals Office Information section if any part of the appeals address is populated.

RSD 2.6.5.11 Enhance VistA (IB) - The Insurance Company Entry/Edit option shall display the address associated with appeals under the Appeals Office Information section if any part of the appeals address is populated.

RSD 2.6.6.1 Enhance VistA (ICB) - ICB shall verify the user has the proper security keys before allowing a new insurance company to be added.

RSD 2.6.6.2 Enhance VistA (ICB) - ICB shall verify the user has the proper security keys before allowing a new group plan to be added while ensuring that the new group plan is not a duplicate.

RSD 2.6.7.1 Enhance VistA (HPID) - VistA shall send a HPID query to the NIF interface when the TRANSMIT ELECTRONICALLY field (#3.01) in the Insurance Company file (#36) is changed to ‘YES-LIVE'.

RSD 2.6.7.2 Enhance VistA (HPID) - The current trigger in Vista to send an HPID query upon modification of either the Professional or Institutional EDI Payer IDs within the Insurance Company file (#36) shall be modified to only allow the creation of the NIF query when the EDI Transmit flag is set to 'YES-LIVE'.

RSD 2.6.7.3 Enhance VistA (HPID) - The current trigger in Vista to send an HPID query upon modification of either the Professional or Institutional EDI Payer IDs within the Insurance Company file (#36) shall be modified to only allow the creation of the NIF query when the Type of Coverage field (#.13) is something other than: Tort/Feasor, Prescription Only, Medicaid, Medi-Cal, Disability Income Insurance, Worker's Compensation, VA Special Class.

RSD 2.6.7.4 Enhance VistA (HPID) - VistA shall be modified to check the existing “IB NIF TCP” entry in the HL Logical Link file (#870) on a daily basis and inform a client defined mailgroup if the logical link is not able to successfully send an HL7 message.

RSD 2.6.7.5 Enhance VistA (HPID) - VistA shall send a HPID query to the NIF interface when the TYPE OF COVERAGE field (#.13) in the Insurance Company file (#36) is changed to an allowable coverage type (see other requirements) and all other rules are met (i.e. TRANSMIT ELECTRONICALLY field (#36,.3.01) is equal to “YES-LIVE”).

RSD 2.6.8.1 Enhance VistA (DD) – VistA shall be modified to include new value/entry in the data dictionary for TYPE OF COVERAGE (#355.2) = VA SPECIAL CLASS. This can be user selected from the following path: Patient Insurance Menu option (PI), followed by Insurance Company Entry/Edit option (EI), followed by Billing/EDI Param action (BP).

RSD 2.6.8.2 Enhance VistA (DD) – VistA shall be modified to include new value/entry in the data dictionary for TYPE OF PLAN (#355.1) = VA SPECIAL CLASS. This can be user selected from the following two paths. Path: Patient Insurance Menu option (PI), followed by Insurance Company Entry/Edit option (EI), followed by View Plans action (VP), followed by View/Edit Plan action (VP), followed by Change Plan Info action (PI). Path: Patient Insurance Menu option (PI), followed by Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Change Plan Info action (PI).

#### Design Element Tables (Enhance VistA)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNHUT2 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.7.4 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBAMTC  IBCNHHLI  IBCNHHLO  IBCNHSRV  IBCNHUT1 | | | | | SEND^IBCNHHLO  HPD^IBCNHUT1  NIF^IBCNHUT1  PHP^IBCNHUT1  SHP^IBCNHUT1  STAT^IBCNHUT1  UID^IBCNHUT1  UNSOL^IBCNHUT1  VID^IBCNHUT1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNHUT2 ;ALB/GEF - HPID/OEID UTILITIES ;11-MAR-14.  .   ; R36(INS,DATA) ; this function gathers all the insurance company data we need to send to the NIF  ; INS= ien of insurance company entry (required)   Q 1  .  .  Q 0 | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNHUT2 ;ALB/GEF - HPID/OEID UTILITIES ;11-MAR-14  .  .   ; R36(INS,DATA) ; this function gathers all the insurance company data we need to send to the NIF  ; INS= ien of insurance company entry (required)   Q 1  **; SETUPVER ; 2.6.7.4 Set up verifying of “IB NIF TCP” logical link N IIV,CURRTIME,MTIME,MSG,MGRP  N ZTRTN,ZTDESC,ZTDTH,ZTIO,ZTUCI,ZTCPU,ZTPRI,ZTSAVE,ZTKIL,ZTSYNC,ZTSK  ;  S IIV=$G(???) ; Get IB NIF TCP data  I '$P(IIV,U,3) G SETUPVRX          ; MM message time is not defined  ;  S CURRTIME=$P($H,",",2) ; current $H time  S MTIME=DT\_"."\_$P(IIV,U,3) ; build a FileMan date/time  S MTIME=$$FMTH^XLFDT(MTIME) ; convert to $H format  S MTIME=$P(MTIME,",",2) ; $H time of MM message  ;  ; If the current time is after the MailMan message time, then schedule the message for tomorrow at that time.  ; Otherwise, schedule it for later today.  S ZTDTH=$S(CURRTIME>MTIME:$H+1,1:+$H)\_","\_MTIME  ;  ; Set up the other TaskManager variables  S ZTRTN="VERFYLNK^IBCNHUT2"              ; THE TAG that we want to TASKMAN to call  S ZTDESC="eIV Verification of “”IB NIF TCP”” link"  S ZTIO=""  D ^%ZTLOAD ; Call TaskManager  I $G(ZTSK) G SETUPVRX    ; Task# is OK so get out  ;  ; Send a MailMan message if this Task could not get scheduled  S MSG(1)="TaskManager could not schedule the daily verification of “”IB NIF TCP”” link"  S MSG(2)="at the specified time of "\_$E($P(IIV,U,3),1,2)\_":"\_$E($P(IIV,U,3),3,4)\_"."  ; Get MAILGROUP  S MGRP=[code]  I MGRP'="" D MSG^IBCNEUT5(MGRP," Daily verification of ““IB NIF TCP”” link Not Scheduled","MSG(")  ; SETUPVRX ;  Q  ; VERFYLNK ; 2.6.74 Verify “IB NIF TCP” entry in the HL Logical Link file (#870) on a daily basis  ;  N CACHEFLG,HLQUIET,MGRP,MSG,RTNTAG,X  S CACHEFLG=($$VERSION^%ZOSV(1)[“Cache”)  S RTNTAG="PING^"\_$S(CACHEFLG:"HLMA",1:"HLOPING")  S HLQUIET=1  D @RTNTAG  S X=$P(X,$C(28))  I X’=INPUT(1) D  . S X=$S(X="":"No response",1:"Incorrect response")   . ; Send a MailMan message if PING not correctly acknowledged  . S MSG(1)="Daily verification of “”IB NIF TCP”” was unsuccessful (“\_X\_”)"  . S MSG(2)="at the specified time of "\_$E($P(IIV,U,3),1,2)\_":"\_$E($P(IIV,U,3),3,4)\_"."  . ; Get MAILGROUP  . S MGRP=[code]  . I MGRP'="" D MSG^IBCNEUT5(MGRP," Daily verification of “”IB NIF TCP”” link: "\_X,"MSG(")  Q  ;** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBAMTC | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.7.4 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | IFCVEXP^IBACN  CVEXMAIL^IBACV  ISLTCADM^IBAECN1  NJ^IBAECN1  ^IBAERR1  PURGE^IBAERR3  BULL^IBAMTC1  MAIN^IBAMTC2  ADM^IBAMTI  BJG^IBAMTI  RELPR^IBAMTV3  EN^IBARXEL  UPDT^IBARXEPS  NIGHT^IBARXMA  ^IBATEI1  SITE^IBAUTL  SERV^IBAUTL2  CLOCK^IBAUTL3  CLOCKL^IBAUTL3  CLUPD^IBAUTL3  EVFIND^IBAUTL3  EVUPD^IBAUTL3  ^IBAUTL4  FILER^IBAUTL5  LAST^IBAUTL5  SECT^IBAUTL5  STD^IBAUTL5  SWSTAT^IBBAPI  ^IBCD  EN^IBCE  SENDEII^IBCNFSND  SEND^IBCNHHLO  HPD^IBCNHUT1  NIF^IBCNHUT1  PHP^IBCNHUT1  SHP^IBCNHUT1  STAT^IBCNHUT1  UID^IBCNHUT1  UNSOL^IBCNHUT1  VID^IBCNHUT1  PUR^IBCNHUT2  BJ^IBJDE  EN^IBOHRL  FY^IBOUTL  NIGHTLY^IBTRKR | | | |
| **Data Dictionary (DD) References** |  | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBAMTC ;ALB/CPM-MEANS TEST NIGHTLY COMPILATION JOB ;09-OCT-91  .  .  CLEAN S %H=+$H-1 D YMD^%DTC S IBDT=X,(IBN,DFN)=0,IBWHER=23  .  .   ; purge HPID files -- IB\*2.0\*519  D PUR^IBCNHUT2  ;  .  .  ; Monitor special inpatient billing cases | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBAMTC ;ALB/CPM-MEANS TEST NIGHTLY COMPILATION JOB ;09-OCT-91  .  .  CLEAN S %H=+$H-1 D YMD^%DTC S IBDT=X,(IBN,DFN)=0,IBWHER=23  .  .   ; purge HPID files -- IB\*2.0\*519  D PUR^IBCNHUT2  ;  .  .  **; 2.6.7.4 Set up verify of “IB NIF TCP” entry in the HL Logical Link file (#870) on a daily basis- IB\*2.0\*549  D SETUPVER^IBCNHUT2  ;.**  .  ;  ; Monitor special inpatient billing cases | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSP** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.2.3, 2.6.5.1, 2.6.5.3, 2.6.9.7, 2.6.9.8, 2.6.9.9 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit [IBCN Patient Insurance (PI)] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNSA0  IBCNSA1  IBCNSC  IBCNSC01  IBCNSC02  IBCNS1  IBCNSD  IBCNSJ1  IBCNSJ2  IBCNSJ5  IBCNSP01  IBCNSP1  IBCNSP3  IBJTNA  IBJTNB  IBTRCD  IBTRCD0  IBTRDD  IBTRED  IBTRED0  IBTRED01  IBTRVD  IBTRVD0 | | | | | INS^IBCNSP0  POLICY^IBCNSP0  SUBSC^IBCNSP01  PRV^IBCNSP01  SPON^IBCNSP0  ID^IBCNSP01  VER^IBCNSP01  CONTACT^IBCNSP0  RIDER^IBCNSP01’  DAT1^IBOUTL  EXPAND^IBTRE | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSM VIEW PAT POLICY, IBCNSP EDIT EFFECTIVE DATES | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNSP ;ALB/AAS - INSURANCE MANAGEMENT - EXPANDED POLICY ;05-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*6,28,43,52,85,251,363,371,416,497\*\*;21-MAR-94;Build 120  ;;Per VHA Directive 2004-038, this routine should not be modified. % ; EN ; -- main entry point for IBCNS EXPANDED POLICY  N IB1ST  K VALMQUIT,IBPPOL  S IBTOP="IBCNSP"  D EN^VALM("IBCNS EXPANDED POLICY")  Q  ; HDR ; -- header code  N W,X,Y,Z  S VALMHDR(1)="Expanded Policy Information for: "\_$E($P(^DPT(DFN,0),U),1,20)\_" "\_$P($$PT^IBEFUNC(DFN),U,2)  S Z=$G(^DPT(DFN,.312,+$P(IBPPOL,U,4),0))  S W=$P($G(^IBA(355.3,+$P(Z,U,18),0)),U,11)  S Y=$E($P($G(^DIC(36,+Z,0)),U),1,20)\_" Insurance Company"  S X="\*\* Plan Currently "\_$S(W:"Ina",1:"A")\_"ctive \*\*"  S VALMHDR(2)=$$SETSTR^VALM1(X,Y,48,29)  Q  ; INIT ; -- init variables and list array  K VALMQUIT  S VALMCNT=0,VALMBG=1  I '$D(IBPPOL) D PPOL Q:$D(VALMQUIT)  K ^TMP("IBCNSVP",$J)  D BLD,HDR  Q  ; BLD ; -- list builder  K ^TMP("IBCNSVP",$J),^TMP("IBCNSVPDX",$J)  D KILL^VALM10()  N IBCDFND,IBCDFND1,IBCDFND2,IBCDFND4,IBCDFND5,IBCDFND7  S IBCDFND=$G(^DPT(DFN,.312,$P(IBPPOL,U,4),0)),IBCDFND1=$G(^(1)),IBCDFND2=$G(^(2)),IBCDFND4=$G(^(4)),IBCDFND5=$G(^(5)),IBCDFND7=$G(^(7))  S IBCPOL=+$P(IBCDFND,U,18),IBCNS=+IBCDFND,IBCDFN=$P(IBPPOL,U,4)  S IBCPOLD=$G(^IBA(355.3,+$P(IBCDFND,U,18),0)),IBCPOLD1=$G(^(1))  S IBCPOLD2=$G(^IBA(355.3,+$G(IBCPOL),6)) ;; Daou/EEN adding BIN and PCN  S IBCPOLDL=$G(^IBA(355.3,+$G(IBCPOL),2)) ;IB\*2\*497 new group name and group number locations  ;  D INS^IBCNSP0 ; insurance company  D POLICY^IBCNSP0 ; plan information  D UR                               ; utilization review info  D EFFECT                           ; effective dates & source of info  D SUBSC^IBCNSP01 ; subscriber info  D EMP                              ; subscriber's employer info  D PRV^IBCNSP01 ; subscriber's provider contact info ;IB\*2\*497  D SPON^IBCNSP0 ; insured person's info  D ID^IBCNSP01 ; ins co ID numbers (IB\*2\*371)  D PLIM                             ; plan coverage limitations  D VER^IBCNSP01 ; user/verifier/editor info  D CONTACT^IBCNSP0 ; last insurance contact  D COMMENT                          ; comments - policy & plan  D RIDER^IBCNSP01 ; policy rider info  ;  S VALMCNT=+$O(^TMP("IBCNSVP",$J,""),-1)  Q  ; COMMENT ; -- Comment region  N START,OFFSET,IBL,IBI  S (START,IBL)=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2  S IB1ST("COMMENT")=START  D SET(START,OFFSET," Comment -- Patient Policy ",IORVON,IORVOFF)  S IBL=IBL+1  D SET(IBL,OFFSET,$S($P(IBCDFND1,U,8)="":"None",1:$P(IBCDFND1,U,8)))  S IBL=IBL+1  D SET(IBL,OFFSET," ")  S IBL=IBL+1  D SET(IBL,OFFSET," Comment -- Group Plan ",IORVON,IORVOFF)  S IBI=0 F  S IBI=$O(^IBA(355.3,+IBCPOL,11,IBI)) Q:IBI<1 D  . S IBL=IBL+1  . D SET(IBL,OFFSET," "\_$E($G(^IBA(355.3,+IBCPOL,11,IBI,0)),1,80))  . Q  S IBL=IBL+1 D SET(IBL,OFFSET," ")  S IBL=IBL+1 D SET(IBL,OFFSET," ")  Q  ; EFFECT ; -- Effective date region  N START,OFFSET  S START=$O(^TMP("IBCNSVP",$J,""),-1)-6 ;ib\*2\*497 lines need to be displayed alongside UR region  S OFFSET=45  D SET(START,OFFSET-4," Effective Dates & Source ",IORVON,IORVOFF)  D SET(START+1,OFFSET," Effective Date: "\_$$DAT1^IBOUTL($P(IBCDFND,U,8)))  D SET(START+2,OFFSET,"Expiration Date: "\_$$DAT1^IBOUTL($P(IBCDFND,U,4)))  D SET(START+3,OFFSET," Source of Info: "\_$$EXPAND^IBTRE(2.312,1.09,$P($G(IBCDFND1),U,9)))  D SET(START+4,OFFSET-4,"Policy Not Billable: "\_$S($P($G(^DPT(DFN,.312,IBCDFN,3)),"^",4):"YES",1:"NO"))  Q  ; UR ; -- UR of insurance region  N START,OFFSET  S START=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2 ;IB\*2\*497  D SET(START,OFFSET," Utilization Review Info ",IORVON,IORVOFF)  D SET(START+1,OFFSET," Require UR: "\_$$EXPAND^IBTRE(355.3,.05,$P(IBCPOLD,U,5)))  D SET(START+2,OFFSET," Require Amb Cert: "\_$$EXPAND^IBTRE(355.3,.12,$P(IBCPOLD,U,12)))  D SET(START+3,OFFSET," Require Pre-Cert: "\_$$EXPAND^IBTRE(355.3,.06,$P(IBCPOLD,U,6)))  D SET(START+4,OFFSET," Exclude Pre-Cond: "\_$$EXPAND^IBTRE(355.3,.07,$P(IBCPOLD,U,7)))  D SET(START+5,OFFSET,"Benefits Assignable: "\_$$EXPAND^IBTRE(355.3,.08,$P(IBCPOLD,U,8)))  D SET(START+6,2," ")  Q EMP ; -- Insurance Employer Region   ; ib\*2\*497 move employer lines around  N OFFSET,START,IBADD,COL2  S START=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2  D SET(START,OFFSET," Subscriber's Employer Information ",IORVON,IORVOFF)  D SET(START+1,OFFSET,$$RJ^XLFSTR(" Employment Status: ",20)\_$$EXPAND^IBTRE(2.312,2.11,$P(IBCDFND2,U,11)))  S COL2=START+1  D SET(START+2,OFFSET,$$RJ^XLFSTR("Employer: ",20)\_$P(IBCDFND2,U,9))  D SET(START+3,OFFSET,$$RJ^XLFSTR("Street: ",20)\_$P(IBCDFND2,U,2)) S IBADD=1  I $P(IBCDFND2,U,3)'="" D SET(START+4,OFFSET,$$RJ^XLFSTR("Street 2: ",20)\_$P(IBCDFND2,U,3)) S IBADD=2  I $P(IBCDFND2,U,4)'="" D SET(START+5,OFFSET,$$RJ^XLFSTR("Street 3: ",20)\_$P(IBCDFND2,U,4)) S IBADD=3  D SET(START+3+IBADD,OFFSET,$$RJ^XLFSTR("City/State: ",20)\_$E($P(IBCDFND2,U,5),1,15)\_$S($P(IBCDFND2,U,5)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCDFND2,U,6),0)),U,2)\_" "\_$E($P(IBCDFND2,U,7),1,5))  D SET(START+4+IBADD,OFFSET,$$RJ^XLFSTR("Phone: ",20)\_$P(IBCDFND2,U,8))  D SET(START+5+IBADD,OFFSET," ") ; ib\*2\*497 only 1 blank line to end the section  ;  S START=COL2,OFFSET=40  D SET(START,OFFSET,"Emp Sponsored Plan: "\_$S(+$P(IBCDFND2,U,10):"Yes",1:"No"))  D SET(START+1,OFFSET,"Claims to Employer: "\_$S(+IBCDFND2:"Yes, Send to Employer",1:"No, Send to Insurance Company"))  D SET(START+2,OFFSET," Retirement Date: "\_$$DAT1^IBOUTL($P(IBCDFND2,U,12)))  ; EMPQ Q  ; PLIM ; plan coverage limitations/plan limitation category display  N START,END S START=$O(^TMP("IBCNSVP",$J,""),-1)+1  S IB1ST("PLIM")=START  D LIMBLD^IBCNSC41(START,2)  S END=$O(^TMP("IBCNSVP",$J,""),-1) ; last line constructed  D SET(END+1,2," ") ; 2 blank lines to end this section  D SET(END+2,2," ") PLIMX ;  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K IBPPOL,VALMQUIT,IBCNS,IBCDFN,IBCPOL,IBCPOLD,IBCPOLD1,IBCPOLD2,IBCPOLDL,IBCDFND,IBCDFND1,IBCDFND2,IBVPCLBG,IBVPCLEN  D CLEAN^VALM10,CLEAR^VALM1  Q  ; EXPND ; -- expand code  Q  ; PPOL ; -- select patient, select policy  I '$D(DFN) D  G:$D(VALMQUIT) PPOLQ  .S DIC="^DPT(",DIC(0)="AEQMN" D ^DIC  .S DFN=+Y  I $G(DFN)<1 S VALMQUIT="" G PPOLQ  ;  I '$O(^DPT(DFN,.312,0)) W !!,"Patient doesn't have Insurance" K DFN G PPOL  ;  S DIC="^DPT("\_DFN\_",.312,",DIC(0)="AEQMN",DIC("A")="Select Patient Policy: "  D ^DIC I +Y<1 S VALMQUIT=""  G:$D(VALMQUIT) PPOLQ  S IBPPOL="^2^"\_DFN\_U\_+Y\_U\_$G(^DPT(DFN,.312,+Y,0)) PPOLQ K DIC Q  ; BLANK(LINE) ; -- Build blank line  D SET^VALM10(.LINE,$J("",80))  Q  ; SET(LINE,COL,TEXT,ON,OFF) ; -- set display info in array  I '$D(@VALMAR@(LINE,0)) D BLANK(.LINE) S VALMCNT=$G(VALMCNT)+1  D SET^VALM10(.LINE,$$SETSTR^VALM1(.TEXT,@VALMAR@(LINE,0),.COL,$L(TEXT)))  D:$G(ON)]""!($G(OFF)]"") CNTRL^VALM10(.LINE,.COL,$L(TEXT),$G(ON),$G(OFF))  W:'(LINE#5) "."  Q  ; |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNSP ;ALB/AAS - INSURANCE MANAGEMENT - EXPANDED POLICY ;05-MAR-1993  ;;2.0;INTEGRATED BILLING;\*\*6,28,43,52,85,251,363,371,416,497\*\*;21-MAR-94;Build 120  ;;Per VHA Directive 2004-038, this routine should not be modified. % ; EN ; -- main entry point for IBCNS EXPANDED POLICY  N IB1ST  K VALMQUIT,IBPPOL  S IBTOP="IBCNSP"  D EN^VALM("IBCNS EXPANDED POLICY")  Q  ; HDR ; -- header code  **; All changes in this section 2.6.2.3**   N **DOD**,W,X,Y,Z  S VALMHDR(1)="Expanded Policy Info**~~rmation~~** for: "\_$E($P(^DPT(DFN,0),U),1,20)\_" "\_$P($$PT^IBEFUNC(DFN),U,2) **S DOD=$$GET1^DIQ(2,DFN\_",",.351,"I")  I DOD'="" D  . S DOD=$$FMTE^XLFDT(DOD,"5DZ")**  **. S VALMHDR(1)=VALMHDR(1)\_" DoD: "\_DOD**   S Z=$G(^DPT(DFN,.312,+$P(IBPPOL,U,4),0))  S W=$P($G(^IBA(355.3,+$P(Z,U,18),0)),U,11)  S Y=$E($P($G(^DIC(36,+Z,0)),U),1,20)\_" Insurance Company"  S X="\*\* Plan Currently "\_$S(W:"Ina",1:"A")\_"ctive \*\*"  S VALMHDR(2)=$$SETSTR^VALM1(X,Y,48,29)  Q  ; INIT ; -- init variables and list array  K VALMQUIT  S VALMCNT=0,VALMBG=1  I '$D(IBPPOL) D PPOL Q:$D(VALMQUIT)  K ^TMP("IBCNSVP",$J)  D BLD,HDR  Q  ; BLD ; -- list builder  K ^TMP("IBCNSVP",$J),^TMP("IBCNSVPDX",$J)  D KILL^VALM10()  N IBCDFND,IBCDFND1,IBCDFND2,IBCDFND4,IBCDFND5,IBCDFND7  S IBCDFND=$G(^DPT(DFN,.312,$P(IBPPOL,U,4),0)),IBCDFND1=$G(^(1)),IBCDFND2=$G(^(2)),IBCDFND4=$G(^(4)),IBCDFND5=$G(^(5)),IBCDFND7=$G(^(7))  S IBCPOL=+$P(IBCDFND,U,18),IBCNS=+IBCDFND,IBCDFN=$P(IBPPOL,U,4)  S IBCPOLD=$G(^IBA(355.3,+$P(IBCDFND,U,18),0)),IBCPOLD1=$G(^(1))  S IBCPOLD2=$G(^IBA(355.3,+$G(IBCPOL),6)) ;; Daou/EEN adding BIN and PCN  S IBCPOLDL=$G(^IBA(355.3,+$G(IBCPOL),2)) ;IB\*2\*497 new group name and group number locations  ;  D INS^IBCNSP0 ; insurance company  D POLICY^IBCNSP0 ; plan information  D UR                               ; utilization review info  D EFFECT                           ; effective dates & source of info  D SUBSC^IBCNSP01 ; subscriber info  D EMP                              ; subscriber's employer info  D PRV^IBCNSP01 ; subscriber's provider contact info ;IB\*2\*497  D SPON^IBCNSP0 ; insured person's info  D ID^IBCNSP01 ; ins co ID numbers (IB\*2\*371)  D PLIM                             ; plan coverage limitations  D VER^IBCNSP01 ; user/verifier/editor info **~~D CONTACT^IBCNSP0 ; last insurance contact~~ 2.6.9.9** D COMMENT                          ; comments - policy & plan  D RIDER^IBCNSP01 ; policy rider info  ;  S VALMCNT=+$O(^TMP("IBCNSVP",$J,""),-1)  Q  ; COMMENT ; -- Comment region  N START,OFFSET,IBL,IBI  S (START,IBL)=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2  S IB1ST("COMMENT")=START **; 2.6.9.7 Move group plan comment display here  S IBL=IBL+1  D SET(IBL,OFFSET," Comment -- Group Plan ",IORVON,IORVOFF)  S IBI=0 F  S IBI=$O(^IBA(355.3,+IBCPOL,11,IBI)) Q:IBI<1 D  . S IBL=IBL+1  . D SET(IBL,OFFSET," "\_$E($G(^IBA(355.3,+IBCPOL,11,IBI,0)),1,80))  . Q**  **; 2.6.9.8 Display the two most recent Patient Policy comments**  D SET(START,OFFSET," Comment -- Patient Policy ",IORVON,IORVOFF)  S IBL=IBL+1  D SET(IBL,OFFSET,$S($P(IBCDFND1,U,8)="":"None",1:$P(IBCDFND1,U,8)))  S IBL=IBL+1  D SET(IBL,OFFSET," ") **; 2.6.9.7 Remove group plan comment from here ~~S IBL=IBL+1  D SET(IBL,OFFSET," Comment -- Group Plan ",IORVON,IORVOFF)  S IBI=0 F  S IBI=$O(^IBA(355.3,+IBCPOL,11,IBI)) Q:IBI<1 D  . S IBL=IBL+1  . D SET(IBL,OFFSET," "\_$E($G(^IBA(355.3,+IBCPOL,11,IBI,0)),1,80))  . Q~~**  S IBL=IBL+1 D SET(IBL,OFFSET," ")  S IBL=IBL+1 D SET(IBL,OFFSET," ")  Q  ; EFFECT ; -- Effective date region  N START,OFFSET  S START=$O(^TMP("IBCNSVP",$J,""),-1)-6 ;ib\*2\*497 lines need to be displayed alongside UR region  S OFFSET=45  D SET(START,OFFSET-4," Effective Dates & Source ",IORVON,IORVOFF)  D SET(START+1,OFFSET," Effective Date: "\_$$DAT1^IBOUTL($P(IBCDFND,U,8)))  D SET(START+2,OFFSET,"Expiration Date: "\_$$DAT1^IBOUTL($P(IBCDFND,U,4)))  D SET(START+3,OFFSET," Source of Info: "\_$$EXPAND^IBTRE(2.312,1.09,$P($G(IBCDFND1),U,9)))  **; 2.6.5.1, 2.6.5.3 Change label, display appropriate value**  D SET(START+4,OFFSET-4,"**Is This** Policy **~~Not~~** Billable: "\_$S($P($G(^DPT(DFN,.312,IBCDFN,3)),"^",4):"**YES**",1:"**NO**"))  Q  ; UR ; -- UR of insurance region  N START,OFFSET  S START=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2 ;IB\*2\*497  D SET(START,OFFSET," Utilization Review Info ",IORVON,IORVOFF)  D SET(START+1,OFFSET," Require UR: "\_$$EXPAND^IBTRE(355.3,.05,$P(IBCPOLD,U,5)))  D SET(START+2,OFFSET," Require Amb Cert: "\_$$EXPAND^IBTRE(355.3,.12,$P(IBCPOLD,U,12)))  D SET(START+3,OFFSET," Require Pre-Cert: "\_$$EXPAND^IBTRE(355.3,.06,$P(IBCPOLD,U,6)))  D SET(START+4,OFFSET," Exclude Pre-Cond: "\_$$EXPAND^IBTRE(355.3,.07,$P(IBCPOLD,U,7)))  D SET(START+5,OFFSET,"Benefits Assignable: "\_$$EXPAND^IBTRE(355.3,.08,$P(IBCPOLD,U,8)))  D SET(START+6,2," ")  Q EMP ; -- Insurance Employer Region   ; ib\*2\*497 move employer lines around  N OFFSET,START,IBADD,COL2  S START=$O(^TMP("IBCNSVP",$J,""),-1)+1,OFFSET=2  D SET(START,OFFSET," Subscriber's Employer Information ",IORVON,IORVOFF)  D SET(START+1,OFFSET,$$RJ^XLFSTR(" Employment Status: ",20)\_$$EXPAND^IBTRE(2.312,2.11,$P(IBCDFND2,U,11)))  S COL2=START+1  D SET(START+2,OFFSET,$$RJ^XLFSTR("Employer: ",20)\_$P(IBCDFND2,U,9))  D SET(START+3,OFFSET,$$RJ^XLFSTR("Street: ",20)\_$P(IBCDFND2,U,2)) S IBADD=1  I $P(IBCDFND2,U,3)'="" D SET(START+4,OFFSET,$$RJ^XLFSTR("Street 2: ",20)\_$P(IBCDFND2,U,3)) S IBADD=2  I $P(IBCDFND2,U,4)'="" D SET(START+5,OFFSET,$$RJ^XLFSTR("Street 3: ",20)\_$P(IBCDFND2,U,4)) S IBADD=3  D SET(START+3+IBADD,OFFSET,$$RJ^XLFSTR("City/State: ",20)\_$E($P(IBCDFND2,U,5),1,15)\_$S($P(IBCDFND2,U,5)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCDFND2,U,6),0)),U,2)\_" "\_$E($P(IBCDFND2,U,7),1,5))  D SET(START+4+IBADD,OFFSET,$$RJ^XLFSTR("Phone: ",20)\_$P(IBCDFND2,U,8))  D SET(START+5+IBADD,OFFSET," ") ; ib\*2\*497 only 1 blank line to end the section  ;  S START=COL2,OFFSET=40  D SET(START,OFFSET,"Emp Sponsored Plan: "\_$S(+$P(IBCDFND2,U,10):"Yes",1:"No"))  D SET(START+1,OFFSET,"Claims to Employer: "\_$S(+IBCDFND2:"Yes, Send to Employer",1:"No, Send to Insurance Company"))  D SET(START+2,OFFSET," Retirement Date: "\_$$DAT1^IBOUTL($P(IBCDFND2,U,12)))  ; EMPQ Q  ; PLIM ; plan coverage limitations/plan limitation category display  N START,END S START=$O(^TMP("IBCNSVP",$J,""),-1)+1  S IB1ST("PLIM")=START  D LIMBLD^IBCNSC41(START,2)  S END=$O(^TMP("IBCNSVP",$J,""),-1) ; last line constructed  D SET(END+1,2," ") ; 2 blank lines to end this section  D SET(END+2,2," ") PLIMX ;  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ; EXIT ; -- exit code  K IBPPOL,VALMQUIT,IBCNS,IBCDFN,IBCPOL,IBCPOLD,IBCPOLD1,IBCPOLD2,IBCPOLDL,IBCDFND,IBCDFND1,IBCDFND2,IBVPCLBG,IBVPCLEN  D CLEAN^VALM10,CLEAR^VALM1  Q  ; EXPND ; -- expand code  Q  ; PPOL ; -- select patient, select policy  I '$D(DFN) D  G:$D(VALMQUIT) PPOLQ  .S DIC="^DPT(",DIC(0)="AEQMN" D ^DIC  .S DFN=+Y  I $G(DFN)<1 S VALMQUIT="" G PPOLQ  ;  I '$O(^DPT(DFN,.312,0)) W !!,"Patient doesn't have Insurance" K DFN G PPOL  ;  S DIC="^DPT("\_DFN\_",.312,",DIC(0)="AEQMN",DIC("A")="Select Patient Policy: "  D ^DIC I +Y<1 S VALMQUIT=""  G:$D(VALMQUIT) PPOLQ  S IBPPOL="^2^"\_DFN\_U\_+Y\_U\_$G(^DPT(DFN,.312,+Y,0)) PPOLQ K DIC Q  ; BLANK(LINE) ; -- Build blank line  D SET^VALM10(.LINE,$J("",80))  Q  ; SET(LINE,COL,TEXT,ON,OFF) ; -- set display info in array  I '$D(@VALMAR@(LINE,0)) D BLANK(.LINE) S VALMCNT=$G(VALMCNT)+1  D SET^VALM10(.LINE,$$SETSTR^VALM1(.TEXT,@VALMAR@(LINE,0),.COL,$L(TEXT)))  D:$G(ON)]""!($G(OFF)]"") CNTRL^VALM10(.LINE,.COL,$L(TEXT),$G(ON),$G(OFF))  W:'(LINE#5) "."  Q  ; |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSC0** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.5.4, 2.6.5.5, 2.6.5.6, 2.6.5.7, 2.6.5.8, 2.6.5.9, 2.6.5.10, 2.6.5.11 | | | | | | | | |
| **Related Options** | View Insurance Company (EI) [IBCN VIEW INSURANCE CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCEF21  IBCNSC  IBCSNC1 | | | | | SET^IBCNSP | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0; INTEGRATED BILLING ;\*\*371\*\*; 21-MAR-94;Build 57  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  S START=27,OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  S IBCNS12=$$ADDRESS(IBCNS,.12,5)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  S START=34,OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  S IBCNS16=$$ADDRESS(IBCNS,.16,6)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ; ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry)  ;  I $P($G(^DIC(36,+INS,+NODE)),"^",5) S IBX=$G(^DIC(36,+INS,+NODE)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH),IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7) I INSSAVE'=INS G REDO  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNSC0 ;ALB/NLR - INSURANCE COMPANY EDIT - ;12-MAR-1993  ;;2.0; INTEGRATED BILLING ;\*\*371\*\*; 21-MAR-94;Build 57  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; CLAIMS1 ; display Inpatient Claims information  N OFFSET,START,IBCNS12,IBADD  S START=27,OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Inpatient Claims Office Information ",IORVON,IORVOFF)  S IBCNS12=$$ADDRESS(IBCNS,.12,5)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS12,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS12,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS12,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS12,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS12,"^",4),1,15)\_$S($P(IBCNS12,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS12,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS12,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS12,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS12,"^",9))  Q  ; R1Q Q CLAIMS2 ; display Outpatient Claims information  ;  N OFFSET,START,IBCNS16,IBADD  S START=34,OFFSET=2  D SET^IBCNSP(START,OFFSET+20," Outpatient Claims Office Information ",IORVON,IORVOFF)  S IBCNS16=$$ADDRESS(IBCNS,.16,6)  D SET^IBCNSP(START+1,OFFSET," Company Name: "\_$P($G(^DIC(36,+$P(IBCNS16,"^",7),0)),"^",1))  D SET^IBCNSP(START+2,OFFSET," Street: "\_$P(IBCNS16,"^",1))  D SET^IBCNSP(START+3,OFFSET," Street 2: "\_$P(IBCNS16,"^",2))  N OFFSET S OFFSET=45  D SET^IBCNSP(START+1,OFFSET," Street 3: "\_$P(IBCNS16,"^",3)) S IBADD=1  D SET^IBCNSP(START+1+IBADD,OFFSET," City/State: "\_$E($P(IBCNS16,"^",4),1,15)\_$S($P(IBCNS16,"^",4)="":"",1:", ")\_$P($G(^DIC(5,+$P(IBCNS16,"^",5),0)),"^",2)\_" "\_$E($P(IBCNS16,"^",6),1,5))  D SET^IBCNSP(START+2+IBADD,OFFSET," Phone: "\_$P(IBCNS16,"^",8))  D SET^IBCNSP(START+3+IBADD,OFFSET," Fax: "\_$P(IBCNS16,"^",9))  Q  ; ADDRESS(INS,NODE,PH) ; -- generic find address  ;  N IBX,INSSAVE,IBPH,IBFX,IBCNT,IBA  S IBX="" ;S IBPH="",IBFX="",IBA=""  ; REDO ; gather insurance carrier's main address information   S IBX=$G(^DIC(36,+INS,.11)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",1),IBFX=$P(IBX,"^",9)  ;S IBCNT=$G(IBCNT)+1  ;  ; -- if process the same co. more than once you are in an infinite loop  ;I $D(IBCNT(IBCNS)) G ADDREQ  ;S IBCNT(IBCNS)=""  ;  ; -- gather address information from specific office (Claims, Appeals, Inquiry)  ;  **; 2.6.5.4, 2.6.5.5, 2.6.5.6, 2.6.5.7, 2.6.5.8, 2.6.5.9, 2.6.5.10, 2.6.5.11**  **; Change to collect Address information if ANY of the address fields are populated**  **I $P($G(^DIC(36,+INS,+NODE)),"^",5) S IBX=$G(^DIC(36,+INS,+NODE)),IBPH=$P($G(^DIC(36,+INS,.13)),"^",PH),IBFX=$P($G(IBX),"^",9)  I $P($G(^DIC(36,+INS,+NODE)),"^",7) S INSSAVE=INS,INS=$P($G(^DIC(36,+INS,+NODE)),"^",7)** I INSSAVE'=INS G REDO  ; ADDRESQ ; concatenate company name, address, phone and fax   S $P(IBA,"^",1,6)=$P($G(IBX),"^",1,6)  S $P(IBA,"^",7)=INS  S $P(IBA,"^",8)=IBPH  S $P(IBA,"^",9)=IBFX ADDREQ Q IBA |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSJ5 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.3.1, 2.6.3.2, 2.6.3.3, 2.6.3.9, 2.6.3.11 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit (PI) [IBCN PATIENT INSURANCE], View Insurance Company [IBCN VIEW INSURANCE CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | INIT^IBCNSC4  INACT^IBCNSJ1  INIT^IBCNSU2  HDR^IBCNSC41  LOCKED^IBTRCD1  EDCOV^IBCNSJ51  BLD^IBCNSP | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSC PLAN LIST, IBCNSC PLAN DETAIL, IBCNSP POLICY MENU, IBCNSJ EDIT COVERAGE LIMITS, IBCNSP ANNUAL BENEFITS, IBCNSJ PLAN COMMENT | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .   ; AB ; -- Edit Annual Benefits from insurance company edit OR plan detail edit I $D(IBCPOL) D FULL^VALM1,EN^VALM("IBCNS ANNUAL BENEFITS") S VALMBCK="R" G ABQ  D FULL^VALM1 .  . IA ; -- (In)activate plan from insurance company edit OR plan detail edit  I '$D(^XUSEC("IB INSURANCE SUPERVISOR",DUZ)) W !!,"Sorry, but you do not have the required privileges to inactivate plans." D PAUSE^VALM1 G IAQ  D FULL^VALM1 .  .   ;  PC ; Plan comments  W !!,"You may now enter comments about this plan."  L +^IBA(355.3,+IBCPOL):5 I '$T D LOCKED^IBTRCD1 G PCQ.  .  .   ; CV ;Edit coverage limitations from edit patient policy  D EDCOV^IBCNSJ51  D BLD^IBCNSP  Q  ;  CV1 ;Edit coverage limitations from edit plan  D EDCOV^IBCNSJ51  D INIT^IBCNSC4  Q  ;  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .   ; AB ; -- Edit Annual Benefits from insurance company edit OR plan detail edit **; 2.6.3.3 Add the following security key check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7,"Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**  ;  I $D(IBCPOL) D FULL^VALM1,EN^VALM("IBCNS ANNUAL BENEFITS") S VALMBCK="R" G ABQ  D FULL^VALM1  .  . IA ; -- (In)activate plan from insurance company edit OR plan detail edit  I '$D(^XUSEC("IB INSURANCE SUPERVISOR",DUZ)) W !!,"Sorry, but you do not have the required privileges to inactivate plans." D PAUSE^VALM1 G IAQ **; 2.6.3.9  Add the following security key check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   D FULL^VALM1 .  .  PC ; Plan comments **; 2.6.3.11  Add the following security key check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   W !!,"You may now enter comments about this plan."  L +^IBA(355.3,+IBCPOL):5 I '$T D LOCKED^IBTRCD1 G PCQ.  .  .   ; CV ;Edit coverage limitations from edit patient policy **; 2.6.3.2  Add the following security key check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**  D EDCOV^IBCNSJ51  D BLD^IBCNSP  Q  ;  CV1 ;Edit coverage limitations from edit plan  **;2.6.3.1 Add the following security key check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   D EDCOV^IBCNSJ51  D INIT^IBCNSC4  Q  ;  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSJ1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.3.10 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit (PI) [IBCN PATIENT INSURANCE] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | SUBS^IBCNSJ  DEL^IBCNSJ11  MAIL^IBCNSJ11 NOTACT^IBCNSJ11  GETPL^IBCNSJ12  MSG^IBCNSJ12  NAC^IBCNSJ12  BU^IBCNSJ13  REP^IBCNSJ13  SEL^IBCNSJ14  NOPL^IBCNSJ2  ASK^IBCNSJ4  EN^IBCNSJ4  PLAN^IBCNSM32  BLD^IBCNSP  HDR^IBCNSP | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSJ INACTIVATE PLAN | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .   ; IA ; 'Inactivate Plan' Action  ; Required variable input:  ; DFN -- Pointer to the patient in file #2  ; IBPPOL -- Patient insurance policy definition  ;  D FULL^VALM1.  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .   ; IA ; 'Inactivate Plan' Action  ; Required variable input:  ; DFN -- Pointer to the patient in file #2  ; IBPPOL -- Patient insurance policy definition  ;  **; 2.6.3.10, 2.3.6.11 Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSM1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.3.4 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit (PI) [IBCN PATIENT INSURANCE] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNSP2 | | | | | FEE^IBCNBME  PREG^IBCNBME  REG^IBCNBME  DISP^IBCNS  INSURED^IBCNS1  IX^IBCNS2  PAT^IBCNSEH  POL^IBCNSEH  SEL^IBCNSEH  IBCNSEVT  AFTER^IBCNSEVT  BEFORE^IBCNSEVT  NEW^IBCNSJ3  DP1^IBCNSM1  EDPOL^IBCNSM3  LK^IBCNSM31  COVERED^IBCNSM31  PATPOL^IBCNSM32  DUPCO^IBCNSOK1  WNRBILL^IBEFUNC | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSA ANNUAL BENEFITS, IBCNSM UPDATE ANNUAL BENEFITS | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .   ;  ; AB ; -- Edit Annual Benefits  D FULL^VALM1  N I,J,IBXX,VALMY  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F  S IBXX=$O(VALMY(IBXX)) Q:'IBXX  D  .S IBPPOL=$G(^TMP("IBNSMDX",$J,$O(^TMP("IBNSM",$J,"IDX",IBXX,0))))  .Q:IBPPOL=""  .S IBCNS=$P(IBPPOL,"^",5),IBCPOL=$P(IBPPOL,"^",22)  .D FULL^VALM1  .D EN^VALM("IBCNS ANNUAL BENEFITS")  .Q  S VALMBCK="R" Q  ;  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  .   ; AB ; -- Edit Annual Benefits **;2.6.3.4 – Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**  ; D FULL^VALM1  N I,J,IBXX,VALMY  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F  S IBXX=$O(VALMY(IBXX)) Q:'IBXX  D  .S IBPPOL=$G(^TMP("IBNSMDX",$J,$O(^TMP("IBNSM",$J,"IDX",IBXX,0))))  .Q:IBPPOL=""  .S IBCNS=$P(IBPPOL,"^",5),IBCPOL=$P(IBPPOL,"^",22)  .D FULL^VALM1  .D EN^VALM("IBCNS ANNUAL BENEFITS")  .Q  S VALMBCK="R" Q ; .  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSP11 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.3.5, 2.6.3.6 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit (PI) [IBCN PATIENT INSURANCE] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | INIT^IBCNSC4  POL^IBCNSEH  SUBS^IBCNSJ  BLD^IBCNSP  COMP^IBCNSP3  SAVE^IBCNSP3  UPDATE^IBCNSP3  UPDATPT^IBCNSP3  CHIP^IBCNSU  LOCKED^IBTRCD1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSJ EDIT PLAN INFO, , IBCNSP EDIT ALL | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .   ; PI ; -- edit plan information from policy edit  D FULL^VALM1  N IBCDFN,IBCPOL  S IBCDFN=$P($G(IBPPOL),"^",4)  ;  ; - build a plan on the fly if there is not one present  S IBCPOL=$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)  I IBCPOL="" S IBCPOL=$$CHIP^IBCNSU($G(^DPT(DFN,.312,IBCDFN,0))) I IBCPOL D  ;Stuff in file  .S DIE="^DPT("\_DFN\_",.312,",DR=".18////"\_IBCPOL  .S DA=IBCDFN,DA(1)=DFN  .D ^DIE  .K DA,DR,DIE,DIC  .Q  D PIEDIT(IBCPOL,DFN,IBCDFN)  Q  ;  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  .   ; PI ; -- edit plan information from policy edit  **; 2.6.3.5, 2.6.3.6 Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**  ;   D FULL^VALM1  N IBCDFN,IBCPOL  S IBCDFN=$P($G(IBPPOL),"^",4)  ;  ; - build a plan on the fly if there is not one present  S IBCPOL=$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)  I IBCPOL="" S IBCPOL=$$CHIP^IBCNSU($G(^DPT(DFN,.312,IBCDFN,0))) I IBCPOL D  ;Stuff in file  .S DIE="^DPT("\_DFN\_",.312,",DR=".18////"\_IBCPOL  .S DA=IBCDFN,DA(1)=DFN  .D ^DIE  .K DA,DR,DIE,DIC  .Q  D PIEDIT(IBCPOL,DFN,IBCDFN)  Q  ;  .  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSP1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.3.7, 2.6.3.8, 2.6.3.12 | | | | | | | | |
| **Related Options** | Patient Insurance Info View\Edit (PI) [IBCN PATIENT INSURANCE], View Insurance Company [IBCN View Insurance Co] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCEF73  IBCE0B01  IBCNSM3  IBCNSM31  IBCNSM32  IBCNUPD | | | | | SCRUB^IBCEF21  INIT^IBCNSC4  PAT^IBCNSEH  POL^IBCNSEH  IBCNSEVT  AFTER^IBCNSEVT  BEFORE^IBCNSEVT  VFY^IBCNSM2  EDPOL^IBCNSM3  PATPOL^IBCNSM32  BLD^IBCNSP  EN^IBCNSP  AI^IBCNSP02  AIP^IBCNSP02  UPDCLM^IBCNSP2  COMP^IBCNSP3  COMPPT^IBCNSP3  SAVE^IBCNSP3  SAVEPT^IBCNSP3  UPDATE^IBCNSP3  UPDATPT^IBCNSP3 VARS^IBCNSP3  LOCKED^IBTRCD1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSJ PLAN UR INFO, IBCNSP UR INFO | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .   ; EA ; -- Edit all  N IBCDFN,IBTRC,IBTRN  D FULL^VALM1 W !!  .  .   ; IT ; -- edit insurance type info from patient policy and plan edit  D FULL^VALM1 W !!  N IBCDFN  S IBCDFN=+$P($G(IBPPOL),"^",4),IBCPOL=+$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)  I 'IBCPOL W !!,"Can't identify the plan!" S VALMBCK="" G ITQ  D ITEDIT(IBCPOL,IBCDFN) ITQ S VALMBCK="R" Q  ;  IT1 ; -- edit insurance type info from patient policy  D ITEDIT(IBCPOL)  S VALMBCK="R"  Q   ;  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .   ; EA ; -- Edit all  N IBCDFN,IBTRC,IBTRN **; 2.6.3.12 Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   D FULL^VALM1 W !!  .  .   ; IT ; -- edit insurance type info from patient policy and plan edit **; 2.6.3.8 Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   D FULL^VALM1 W !!  N IBCDFN  S IBCDFN=+$P($G(IBPPOL),"^",4),IBCPOL=+$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)  I 'IBCPOL W !!,"Can't identify the plan!" S VALMBCK="" G ITQ  D ITEDIT(IBCPOL,IBCDFN) ITQ S VALMBCK="R" Q  ;  IT1 ; -- edit insurance type info from patient policy **; 2.6.3.7 Add the following security check**  **I '$D(^XUSEC("IB GROUP PLAN EDIT",DUZ)) D  Q  . W !,\*7," Invalid Security to perform this action."  . S DLINE=""  . K DIR  . D PAUSE^VALM1**   D ITEDIT(IBCPOL)  S VALMBCK="R"  Q  ;  .  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNES** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.4.1, 2.6.4.3, 2.6.4.5, 2.6.4.6 | | | | | | | | |
| **Related Options** | Process Insurance Buffer [IBCN INSURANCE BUFFER PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBCD  IBCNBLB  IBCNERPE  IBCNES2  IBCNS3 | | | | | SERVLN^IBCNBLE  BRE^IBCNBLE  CMPI^IBCNES1  EB^IBCNES1  HCSD^IBCNES1  NTE^IBCNES1  SET^IBCNES1  RPDM^IBCNES3  FO^IBCNEUT1  PT^IBEFUNC | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNB PROCESS ACCEPT | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .  ; INIT(IBVF,IBVIENS,IBVEBFLG,IBVV,IBVSUB) ; List Entry  ;  ; IBVF = file# 2.322 or 365.02 (required)  ; IBVIENS = std IENS list of internal entry numbers - NOT including any EB iens (required)  ; IBVEBFLG = flag indicating which EB records to pull  ; "A" - all of them  ; "L" - only the last one (default)  ; "F" - only the first one  ; "M" - multiple, pass IBEBFLG by reference and include the IB iens in  ; an array as follows:  ; IBVEBFLG="M"  ; IBVEBFLG(3)=""  ; IBVEBFLG(5)=""  ; IBVV = Video attributes flag  ; 1 = reverse video (default)  ; 2 = bold  ; 3 = underline  ; IBVSUB = literal subscript to use in the display scratch global  ;  N IBVDA,GLO,IBVLIST,IEN,IBVEBIEN,IBVEBTOT,IBVEBCNT  N IBECODE,IIVSTAT,PLNDESC,IBINSTYP,OTHINS,MWNRIEN     ;IB\*2.0\*506  ;  S OTHINS=0 ;IB\*2.0\*506/TAZ Initialize Other Insurance variable  S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25) ;IB\*2.0\*506/TAZ Initialize Medicare WNR payer IEN  S IBVSUB=$G(IBVSUB)  I IBVSUB="" S IBVSUB="EB ELIG/BEN"  K ^TMP(IBVSUB,$J)  I $D(VALMEVL) D CLEAN^VALM10,KILL^VALM10()  ;  D DA^DILF(IBVIENS,.IBVDA) ; build the IBVDA array for the iens  I '$D(IBVDA) D NODATA G INITX  ;  I $D(VALMEVL),'$G(IBVV) S IBVV=1 ; default reverse video for ListMan  I '$D(VALMEVL) S IBVV=""            ; no video attributes for non-ListMan  ;  D RPDM^IBCNES3($S(IBVF=365.02:365,1:2.312),.IBVDA,IBVV,IBVSUB) ; IB\*2\*497 display group level eligibility information  ;  I IBVF=2.322 S GLO=$NA(^DPT(+$G(IBVDA(1)),.312,+$G(IBVDA),6)) ; pt. insurance  I IBVF=365.02 S GLO=$NA(^IBCN(365,+$G(IBVDA),2)) ; response file  I $G(GLO)="" D NODATA G INITX  ;  S IBVEBFLG=$G(IBVEBFLG,"L")  K IBVLIST  I IBVEBFLG="L" S IEN=+$O(@GLO@(" "),-1) I IEN S IBVLIST(IEN)=""            ; last EB ien on file  I IBVEBFLG="F" S IEN=+$O(@GLO@(0)) I IEN S IBVLIST(IEN)=""                 ; first EB ien on file  I IBVEBFLG="A" S IEN=0 F  S IEN=$O(@GLO@(IEN)) Q:'IEN  S IBVLIST(IEN)=""   ; all EB iens on file  I IBVEBFLG="M" S IEN=0 F  S IEN=$O(IBVEBFLG(IEN)) Q:'IEN  I $D(@GLO@(IEN)) S IBVLIST(IEN)=""   ; multiple  ;  I '$D(IBVLIST) D NODATA G INITX  ;  ; count them  S IEN=0 F IBVEBTOT=0:1 S IEN=$O(IBVLIST(IEN)) Q:'IEN  I 'IBVEBTOT D NODATA G INITX  ;  ; /IB\*2.0\*506 Beginning  ; Count EBs and gather EB Summary Data  ; IIVSTAT will tell us the coverage status 1,6, or V (File #365.011)  ; Flag related to IBINSTYP will tell us the insurance type (File #365.014)  ; OTHINS will tell us if Other Insurance was indicated on the response  ;  S (IEN,IBVEBTOT,OTHINS)=0,(IIVSTAT,IBINSTYP,PLNDESC)=""  F  S IEN=$O(IBVLIST(IEN)) D  Q:'IEN  . Q:'IEN  . S IBVEBTOT=IBVEBTOT+1 ; total # of EBs  . I IBVEBTOT=1 D  . . S IBECODE=$P($G(@GLO@(1,0)),U,2) ; Eligibility/Benefits Code  . . S PLNDESC=$P($G(@GLO@(1,0)),U,6) ; Plan Description  . . I PLNDESC'="eIV Eligibility Determination" S IIVSTAT="V"  . . I IBECODE=1 S IIVSTAT=1 ; active  . . I IBECODE=6 S IIVSTAT=6 ; inactive  . . I IIVSTAT="" S IIVSTAT="V"           ; ambigious  . . ;  . I IBINSTYP="" D  . . S IBINSTYP=$P($G(@GLO@(IEN,0)),U,5) ; Insurance Type (check all EBs, get 1st occurrence)  . . I IBINSTYP="" Q   ; no insurance type found   . . S IBINSTYP=$$GET1^DIQ(365.014,IBINSTYP,.02)  . ;  . ;Screen out non\_Medicare records  . S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25) ; Initialize Medicare WNR payer IEN  . I IBVF=2.322,($$GET1^DIQ(36,$P(^DPT(+$G(IBVDA(1)),.312,+$G(IBVDA),0),U,1)\_",",3.1,"I")'=MWNRIEN) Q  . I IBVF=365.02,($P(^IBCN(365,+$G(IBVDA),0),U,3)'=MWNRIEN) Q  . ;  . N IBEIEN,IBELIG  . S IBEIEN=0  . F  S IBEIEN=$O(@GLO@(IBEIEN)) Q:'IBEIEN  D  I OTHINS Q  .. ;Get Eligibility Code. We want R codes only.  .. S IBELIG=$P($G(@GLO@(IBEIEN,0)),U,2) I $P($G(^IBE(365.011,IBELIG,0)),U,1)="R" S OTHINS=1  ;  I IBVEBTOT D SUMMARY(IIVSTAT,IBINSTYP,OTHINS)  ; /IB\*2.0\*506 End  ;  I 'IBVEBTOT D NODATA G INITX  ;  S (IBVEBIEN,IBVEBCNT)=0  F  S IBVEBIEN=$O(IBVLIST(IBVEBIEN)) Q:'IBVEBIEN  D  . S IBVEBCNT=IBVEBCNT+1  . N TXVIENS  . ;  . ; if there is more than 1 EB group, then display a header line for separation  . I IBVEBTOT>1 D  .. N DSP,LN,IBZ  .. S DSP=$NA(^TMP(IBVSUB,$J,"DISP"))  .. S LN=+$O(@DSP@(""),-1)  .. S IBZ="eIV Eligibility/Benefit Data Group# "\_IBVEBCNT\_" of "\_IBVEBTOT  .. S IBZ=$$FO^IBCNEUT1($J("",20)\_IBZ,80)  .. S LN=LN+1 D SET^IBCNES1(LN,1,IBZ,,IBVV)  .. S LN=LN+1 D SET^IBCNES1(LN)  .. Q  . ;  . ; add this EB ien to the list of iens  . S TXVIENS=IBVEBIEN\_","\_IBVIENS  . ;  . ; call the screen sections to build the display  . D EB^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D CMPI^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D HCSD^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D NTE^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D BRE^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . ;  . Q  ;  S VALMCNT=$O(^TMP(IBVSUB,$J,"DISP"," "),-1)  ; INITX ;  Q  ;  .  .  . |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  .   ; INIT(IBVF,IBVIENS,IBVEBFLG,IBVV,IBVSUB) ; List Entry  ;  ; IBVF = file# 2.322 or 365.02 (required)  ; IBVIENS = std IENS list of internal entry numbers - NOT including any EB iens (required)  ; IBVEBFLG = flag indicating which EB records to pull  ; "A" - all of them  ; "L" - only the last one (default)  ; "F" - only the first one  ; "M" - multiple, pass IBEBFLG by reference and include the IB iens in  ; an array as follows:  ; IBVEBFLG="M"  ; IBVEBFLG(3)=""  ; IBVEBFLG(5)=""  ; IBVV = Video attributes flag  ; 1 = reverse video (default)  ; 2 = bold  ; 3 = underline  ; IBVSUB = literal subscript to use in the display scratch global  ;  N IBVDA,GLO,IBVLIST,IEN,IBVEBIEN,IBVEBTOT,IBVEBCNT  N IBECODE,IIVSTAT,PLNDESC,IBINSTYP,OTHINS,MWNRIEN     ;IB\*2.0\*506  ;  S OTHINS=0 ;IB\*2.0\*506/TAZ Initialize Other Insurance variable  S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25) ;IB\*2.0\*506/TAZ Initialize Medicare WNR payer IEN  S IBVSUB=$G(IBVSUB)  I IBVSUB="" S IBVSUB="EB ELIG/BEN"  K ^TMP(IBVSUB,$J)  I $D(VALMEVL) D CLEAN^VALM10,KILL^VALM10()  ;  D DA^DILF(IBVIENS,.IBVDA) ; build the IBVDA array for the iens  I '$D(IBVDA) D NODATA G INITX  ;  I $D(VALMEVL),'$G(IBVV) S IBVV=1 ; default reverse video for ListMan  I '$D(VALMEVL) S IBVV=""            ; no video attributes for non-ListMan  **;2.6.4.1, 2.6.4.3, 2.6.4.5, 2.6.4.6**  **; NEED TO ADD CODE HERE THAT WILL SET A SWITCH TO SAY THAT WE ARE**  **; BEING CALLED FROM THE “EB” ACTION OF THE INSURANCE BUFFER.**  **; THEN WE NEED TO CALL OUT AND OBTAIN THE RESPONSE REPORT PORTION**  **; TO BE DISPLAYED BEFORE THE “ELIGIBILITY/GROUP PLAN INFORMATION” SECTION.**  **🡺N EBSWT S EBSWT=1**  **🡺D DATA^IBCNERPE(DISPDATA,EBSWT) ; IB\*2\*549 display the response report section**  ;  D RPDM^IBCNES3($S(IBVF=365.02:365,1:2.312),.IBVDA,IBVV,IBVSUB) ; IB\*2\*497 display group level eligibility information  ;  I IBVF=2.322 S GLO=$NA(^DPT(+$G(IBVDA(1)),.312,+$G(IBVDA),6)) ; pt. insurance  I IBVF=365.02 S GLO=$NA(^IBCN(365,+$G(IBVDA),2)) ; response file  I $G(GLO)="" D NODATA G INITX  ;  S IBVEBFLG=$G(IBVEBFLG,"L")  K IBVLIST  I IBVEBFLG="L" S IEN=+$O(@GLO@(" "),-1) I IEN S IBVLIST(IEN)=""            ; last EB ien on file  I IBVEBFLG="F" S IEN=+$O(@GLO@(0)) I IEN S IBVLIST(IEN)=""                 ; first EB ien on file  I IBVEBFLG="A" S IEN=0 F  S IEN=$O(@GLO@(IEN)) Q:'IEN  S IBVLIST(IEN)=""   ; all EB iens on file  I IBVEBFLG="M" S IEN=0 F  S IEN=$O(IBVEBFLG(IEN)) Q:'IEN  I $D(@GLO@(IEN)) S IBVLIST(IEN)=""   ; multiple  ;  I '$D(IBVLIST) D NODATA G INITX  ;  ; count them  S IEN=0 F IBVEBTOT=0:1 S IEN=$O(IBVLIST(IEN)) Q:'IEN  I 'IBVEBTOT D NODATA G INITX  ;  ; /IB\*2.0\*506 Beginning  ; Count EBs and gather EB Summary Data  ; IIVSTAT will tell us the coverage status 1,6, or V (File #365.011)  ; Flag related to IBINSTYP will tell us the insurance type (File #365.014)  ; OTHINS will tell us if Other Insurance was indicated on the response  ;  S (IEN,IBVEBTOT,OTHINS)=0,(IIVSTAT,IBINSTYP,PLNDESC)=""  F  S IEN=$O(IBVLIST(IEN)) D  Q:'IEN  . Q:'IEN  . S IBVEBTOT=IBVEBTOT+1 ; total # of EBs  . I IBVEBTOT=1 D  . . S IBECODE=$P($G(@GLO@(1,0)),U,2) ; Eligibility/Benefits Code  . . S PLNDESC=$P($G(@GLO@(1,0)),U,6) ; Plan Description  . . I PLNDESC'="eIV Eligibility Determination" S IIVSTAT="V"  . . I IBECODE=1 S IIVSTAT=1 ; active  . . I IBECODE=6 S IIVSTAT=6 ; inactive  . . I IIVSTAT="" S IIVSTAT="V"           ; ambigious  . . ;  . I IBINSTYP="" D  . . S IBINSTYP=$P($G(@GLO@(IEN,0)),U,5) ; Insurance Type (check all EBs, get 1st occurrence)  . . I IBINSTYP="" Q   ; no insurance type found   . . S IBINSTYP=$$GET1^DIQ(365.014,IBINSTYP,.02)  . ;  . ;Screen out non\_Medicare records  . S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25) ; Initialize Medicare WNR payer IEN  . I IBVF=2.322,($$GET1^DIQ(36,$P(^DPT(+$G(IBVDA(1)),.312,+$G(IBVDA),0),U,1)\_",",3.1,"I")'=MWNRIEN) Q  . I IBVF=365.02,($P(^IBCN(365,+$G(IBVDA),0),U,3)'=MWNRIEN) Q  . ;  . N IBEIEN,IBELIG  . S IBEIEN=0  . F  S IBEIEN=$O(@GLO@(IBEIEN)) Q:'IBEIEN  D  I OTHINS Q  .. ;Get Eligibility Code. We want R codes only.  .. S IBELIG=$P($G(@GLO@(IBEIEN,0)),U,2) I $P($G(^IBE(365.011,IBELIG,0)),U,1)="R" S OTHINS=1  ;  I IBVEBTOT D SUMMARY(IIVSTAT,IBINSTYP,OTHINS)  ; /IB\*2.0\*506 End  ;  I 'IBVEBTOT D NODATA G INITX  ;  S (IBVEBIEN,IBVEBCNT)=0  F  S IBVEBIEN=$O(IBVLIST(IBVEBIEN)) Q:'IBVEBIEN  D  . S IBVEBCNT=IBVEBCNT+1  . N TXVIENS  . ;  . ; if there is more than 1 EB group, then display a header line for separation  . I IBVEBTOT>1 D  .. N DSP,LN,IBZ  .. S DSP=$NA(^TMP(IBVSUB,$J,"DISP"))  .. S LN=+$O(@DSP@(""),-1)  .. S IBZ="eIV Eligibility/Benefit Data Group# "\_IBVEBCNT\_" of "\_IBVEBTOT  .. S IBZ=$$FO^IBCNEUT1($J("",20)\_IBZ,80)  .. S LN=LN+1 D SET^IBCNES1(LN,1,IBZ,,IBVV)  .. S LN=LN+1 D SET^IBCNES1(LN)  .. Q  . ;  . ; add this EB ien to the list of iens  . S TXVIENS=IBVEBIEN\_","\_IBVIENS  . ;  . ; call the screen sections to build the display  . D EB^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D CMPI^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D HCSD^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D NTE^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . D BRE^IBCNES1(IBVF,TXVIENS,IBVV,IBVSUB)  . ;  . Q  ;  S VALMCNT=$O(^TMP(IBVSUB,$J,"DISP"," "),-1)  ; INITX ;  Q  ;  .  .  . |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNEHL1** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.4.4, 2.6.4.8, 2.6.4.9 | | | | | | | | |
| **Related Options** | Process Insurance Buffer (BI) [IBCN INSURANCE BUFFER PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | ^IBCNBAR  ^IBCNEHL1 | | | | | PREL^IBCNEHLU  LCKERR^IBCNEHL3  WARN^IBCNEHL3  GRPFILE^IBCNEHL1  EBFILE^IBCNEHL1  UPDIREC^IBCNEHL3  SST^IBCNEUT2  RSTA^IBCNEUT7  STATUS^IBCNBEE  DELDATA^IBCNBED  PYRAPP^IBCNEUT5  ISMCR^IBCNEHLU  ONEPOL^IBCNEHLU  EXPIRED^IBCNEDE2  PATISSUB^IBCNEHLU  CHK1^IBCNEHL3  CHK2^IBCNEHL3 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNB LIST ADD | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNEHL1 ;DAOU/ALA - HL7 Process Incoming RPI Messages ;26-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*300,345,416,444,438,497,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process incoming IIV response messages.  ; This includes updating the record in the IIV Response File,  ; updating the Buffer record (if there is one and creating a new  ; one if there isn't) with the appropriate Buffer Symbol and data  ;  ; Variables  ; SEG = HL7 Segment Name  ; MSGID = Original Message Control ID  ; ACK = Acknowledgment (AA=Accepted, AE=Error)  ; ERTXT = Error Message Text  ; ERFLG = Error quit flag  ; ERACT = Error Action  ; ERCON = Error Condition  ; RIEN = Response Record IEN  ; IIVSTAT = EC generated flag interpreting status of response  ; 1 = + (auto-update requirement)  ; 6 = -  ; V = #  ; MAP = Array that maps EC's IIV status flag to IIV STATUS TABLE (#365.15) IEN  ; EN ; Entry Point  N AUTO,EBDA,ERFLG,ERROR,G2OFLG,HCT,HLCMP,HLREP,HLSCMP,IIVSTAT,IRIEN,MAP,MGRP,RIEN,RSUPDT,SEG,SUBID,TRACE,UP,ACK  S (ERFLG,G2OFLG)=0,MGRP=$$MGRP^IBCNEUT5(),HCT=1,SUBID="",IIVSTAT=""  ;  S HLCMP=$E(HL("ECH")) ; HL7 component separator  S HLSCMP=$E(HL("ECH"),4) ; HL7 subcomponent separator  S HLREP=$E(HL("ECH"),2) ; HL7 repetition separator  ; Create map from EC to VistA  S MAP(1)=8,MAP(6)=9,MAP("V")=21  ;  ; Loop through the message and find each segment for processing  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:ERFLG  .D SPAR^IBCNEHLU  .S SEG=$G(IBSEG(1))  .; check if we are inside G2O group of segments  .I SEG="ZTY" S G2OFLG=1  .I G2OFLG,SEG'="ZTY",SEG'="CTD" S G2OFLG=0  .; If we are outside of Z\_Benefit\_group, kill EB multiple ien  .; I +$G(EBDA),".MSH.MSA.PRD.PID.GT1.IN1.IN3."[("."\_SEG\_".")!('G2OFLG&(SEG="CTD")) K EBDA  .;  .Q:SEG="PRD"  ; IB\*2\*497 PRD segment is not processed  .;  .I SEG="MSA" D MSA^IBCNEHL2(.ERACT,.ERCON,.ERROR,.ERTXT,.IBSEG,MGRP,.RIEN,.TRACE) Q  .;  .; Contact Segment  .I SEG="CTD",'G2OFLG D CTD^IBCNEHL2(.ERROR,.IBSEG,RIEN) Q  .;  .; Patient Segment  .I SEG="PID" D PID^IBCNEHL2(.ERFLG,.ERROR,.IBSEG,RIEN) Q  .;  .; Guarantor Segment  .I SEG="GT1" D GT1^IBCNEHL2(.ERROR,.IBSEG,RIEN,.SUBID) Q  .;  .; Insurance Segment  .I SEG="IN1" D IN1^IBCNEHL2(.ERROR,.IBSEG,RIEN,SUBID) Q  .;  .; Addt'l Insurance Segment  .;I SEG="IN2" ; for future expansion, add IN2 tag to IBCNEHL2  .;  .; Addt'l Insurance - Cert Segment  .I SEG="IN3" D IN3^IBCNEHL2(.ERROR,.IBSEG,RIEN) Q   .;  .; IB\*2\*497 GROUP LEVEL REFERENCE ID segment (x12 loops 2100C and 2100D)  . I SEG="ZRF",'$D(EBDA) D GZRF^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  .;  .; Eligibility/Benefit Segment  .I SEG="ZEB" D ZEB^IBCNEHL2(.EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Healthcare Delivery Segment  .I SEG="ZHS" D ZHS^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit level Reference ID Segment (X12 loops 2110C and 2110D)  .I SEG="ZRF",+$G(EBDA) D ZRF^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  ;IB\*2\*497 add check to make sure z benefit group  .;  .; Subscriber Date Segment  .I SEG="ZSD" D ZSD^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Subscriber Additional Info Segment  .I SEG="ZII" D ZII^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Segment  .I SEG="ZTY" D ZTY^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Contact Segment  .I SEG="CTD",G2OFLG D G2OCTD^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Notes Segment  .I SEG="NTE",+$G(EBDA) D EBNTE^IBCNEHL2(EBDA,.IBSEG,RIEN) Q  .;  .; Reject Reasons Segment  .I SEG="ERR" K ERDA D ERR^IBCNEHL4(.ERDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Notes Segment  .I SEG="NTE",'$D(EBDA),+$G(ERDA) D NTE^IBCNEHL4(ERDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Subscriber date segment (subscriber level)  .I SEG="ZTP" D ZTP^IBCNEHL4(.ERROR,.IBSEG,RIEN) Q  . ; ib\*2\*497 - add processing for ROL, DG1, and ZMP segments  . ; Provider Code segment   . I SEG="ROL" D ROL^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  . ;  . ; Health Care Diagnosis Code segment  . I SEG="DG1" D DG1^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  .;  .; Military Personnel Information segment  . I SEG="ZMP" D ZMP^IBCNEHL5(.ERROR,.IBSEG,RIEN)  ;  S AUTO=$$AUTOUPD(RIEN)  I $G(ACK)'="AE",$G(ERACT)="",$G(ERTXT)="",'$D(ERROR),+AUTO D  Q  .D:$P(AUTO,U,3)'="" AUTOFIL($P(AUTO,U,2),$P(AUTO,U,3),$P(AUTO,U,6))  .D:$P(AUTO,U,4)'="" AUTOFIL($P(AUTO,U,2),$P(AUTO,U,4),$P(AUTO,U,6))  .Q  D FIL  Q  ;  ; ================================================================= AUTOFIL(DFN,IEN312,ISSUB) ; Finish processing the response message - file directly into patient insurance  ;  N BUFF,DATA,ERROR,IENS,PREL,RDATA0,RDATA1,RDATA5,RDATA13,RSTYPE,TQN,TSTAMP,MIL,OKAY   ; IB\*2.0\*497 (vd)  ;  Q:$G(RIEN)=""  S TSTAMP=$$NOW^XLFDT(),IENS=IEN312\_","\_DFN\_","  S RDATA0=$G(^IBCN(365,RIEN,0)),RDATA1=$G(^IBCN(365,RIEN,1)),RDATA5=$G(^IBCN(365,RIEN,5))  S RDATA13=$G(^IBCN(365,RIEN,13)) ; IB\*2.0\*497 (vd)  S TQN=$P(RDATA0,U,5),RSTYPE=$P(RDATA0,U,10)  I ISSUB S DATA(2.312,IENS,7.01)=$P(RDATA13,U) ; name - IB\*2.0\*497 (vd)  S DATA(2.312,IENS,3.01)=$P(RDATA1,U,2) ; dob  S DATA(2.312,IENS,3.05)=$P(RDATA1,U,3) ; ssn  I ISSUB,$P(RDATA1,U,8)'="" S DATA(2.312,IENS,6)=$P(RDATA1,U,8) ; whose insurance  ; pt. relationship (365,8.01) IB\*2\*497 code from 365,8.01 needs evaluation and possible conversion  S PREL=$$GET1^DIQ(365,RIEN,8.01) I ISSUB,PREL'="" S DATA(2.312,IENS,4.03)=$$PREL^IBCNEHLU(2.312,4.03,PREL)  S DATA(2.312,IENS,1.03)=TSTAMP ; date last verified  S DATA(2.312,IENS,1.04)="" ; last verified by  S DATA(2.312,IENS,1.05)=TSTAMP ; date last edited  S DATA(2.312,IENS,1.06)="" ; last edited by  S DATA(2.312,IENS,1.09)=5 ; source of info = eIV  ;subscriber address  S DATA(2.312,IENS,3.06)=$P(RDATA5,U) ; street line 1  S DATA(2.312,IENS,3.07)=$P(RDATA5,U,2) ; street line 2  S DATA(2.312,IENS,3.08)=$P(RDATA5,U,3) ; city  S DATA(2.312,IENS,3.09)=$P(RDATA5,U,4) ; state  S DATA(2.312,IENS,3.1)=$P(RDATA5,U,5) ; zip  S DATA(2.312,IENS,3.13)=$P(RDATA5,U,6) ; country  S DATA(2.312,IENS,3.14)=$P(RDATA5,U,7) ; country subdivision  ;  L +^DPT(DFN,.312,IEN312):15 I '$T D LCKERR^IBCNEHL3 D FIL Q  D FILE^DIE("ET","DATA","ERROR") I $D(ERROR) D WARN^IBCNEHL3 K ERROR D FIL G AUTOFILX  ;  ; set eIV auto-update field separately because of the trigger on field 1.05  K DATA S DATA(2.312,IENS,4.04)="YES" D FILE^DIE("ET","DATA","ERROR") I $D(ERROR) D WARN^IBCNEHL3 G AUTOFILX  S ERFLG=$$GRPFILE(DFN,IEN312,RIEN,1) I $G(ERFLG) G AUTOFILX  ;IB\*2\*497 file data at 2.312, 9, 10 and 11 subfiles; if error is produced update buffer entry and then quit processing  ; file new EB data  S ERFLG=$$EBFILE(DFN,IEN312,RIEN,1)  ; bail out if something went wrong during filing of EB data  I $G(ERFLG) G AUTOFILX  ; update insurance record ien in transmission queue  D UPDIREC^IBCNEHL3(RIEN,IEN312)  ; For an original response, set the Transmission Queue Status to 'Response Received' &  ; update remaining retries to comm failure (5)  I $G(RSTYPE)="O" D SST^IBCNEUT2(TQN,3),RSTA^IBCNEUT7(TQN)  ; update buffer file entry so only stub remains and status is changed  S BUFF=+$P($G(^IBCN(365,RIEN,0)),U,4)  I BUFF D  .D STATUS^IBCNBEE(BUFF,"A",0,0,0) ; update buffer entry's status to accepted  .D DELDATA^IBCNBED(BUFF) ; delete buffer's insurance/patient data  .Q AUTOFILX ;  L -^DPT(DFN,.312,IEN312)  Q  ; GRPFILE(DFN,IEN312,RIEN,AFLG) ; ib\*2\*497 file data at node 12 and at subfiles 2.312,9, 10 and 11  ; DFN - file 2 ien  ; IEN312 - file 2.312 ien  ; RIEN = file 365 ien  ; AFLG - 1 if called from autoupdate, 0 if called from ins. buffer process entry  ; output - returns 0 or 1  ; 0 - entry update received an error when attempting to file  ; 1 - successful update  N DA,Z,Z2,DATA12,IENS,IENS365,IENS312,REF,PROV,DIAG,REF3129,PROV332,DIAG3121,NODE,ERROR,ERFLG  ; retrieve external values of data located at node 12 of 365  S IENS=IEN312\_","\_DFN\_","  D GETS^DIQ(365,RIEN,"12.01:12.07",,"MIL")  M DATA12(2.312,IENS)=MIL(365,RIEN\_",")  D FILE^DIE("ET","DATA12","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; remove existing sub-file entries at nodes 9, 10, and 11 before update of new data  F NODE="9","10","11" D  . S DIK="^DPT("\_DFN\_",.312,"\_IEN312\_","\_NODE\_",",DA(2)=DFN,DA(1)=IEN312  . S DA=0 F  S DA=$O(^DPT(DFN,.312,IEN312,NODE,DA)) Q:DA=""!(DA?1.A) D ^DIK  S IENS312="+1,"\_IEN312\_","\_DFN\_","  ; update node 9 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,9,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,9,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.09,IENS365,"\*",,"REF")  S Z2="" F  S Z2=$O(REF(365.09,Z2)) Q:Z2=""  M REF3129(2.3129,IENS312)=REF(365.09,Z2) D UPDATE^DIE("E","REF3129",,"ERROR") K REF3129 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; update node 10 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,10,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,10,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.04,IENS365,"\*",,"PROV")  S Z2="" F  S Z2=$O(PROV(365.04,Z2)) Q:Z2=""  M PROV332(2.332,IENS312)=PROV(365.04,Z2) D UPDATE^DIE("E","PROV332",,"ERROR") K PROV332 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; update node 11 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,11,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,11,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.01,IENS365,"\*",,"DIAG")  S Z2="" F  S Z2=$O(DIAG(365.01,Z2)) Q:Z2=""  M DIAG3121(2.31211,IENS312)=DIAG(365.01,Z2) D UPDATE^DIE("E","DIAG3121",,"ERROR") K DIAG3121 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR GRPFILEX ;  Q $G(ERFLG)  ; FIL ; Finish processing the response message - file into insurance buffer  ;  ; Input Variables  ; ERACT, ERFLG, ERROR, IIVSTAT, MAP, RIEN, TRACE  ;  ; If no record IEN, quit  I $G(RIEN)="" Q  ;  N BUFF,DFN,FILEIT,IBFDA,IBIEN,IBQFL,RDAT0,RSRVDT,RSTYPE,SYMBOL,TQDATA,TQN,TQSRVDT  ; Initialize variables from the Response File  S RDAT0=$G(^IBCN(365,RIEN,0)),TQN=$P(RDAT0,U,5)  S TQDATA=$G(^IBCN(365.1,TQN,0))  S IBQFL=$P(TQDATA,U,11)  S DFN=$P(RDAT0,U,2),BUFF=$P(RDAT0,U,4)  S IBIEN=$P(TQDATA,U,5),RSTYPE=$P(RDAT0,U,10)  S RSRVDT=$P($G(^IBCN(365,RIEN,1)),U,10)  ;  ; If an unknown error action or an error filing the response message,  ; send a warning email message  ; Note - A call to UEACT will always set ERFLAG=1  ;  ; IB\*2.0\*506 Removed the following line of code to Treat all AAA Action Codes  ; as though the Payer/FSC Responded.  ;I ",W,X,R,P,C,N,Y,S,"'[(","\_$G(ERACT)\_",")&($G(ERACT)'="")!$D(ERROR) D UEACT^IBCNEHL3  ;  ; If an error occurred, processing complete  I $G(ERFLG)=1 Q  ;  ; For an original response, set the Transmission Queue Status to 'Response Received' &  ; update remaining retries to comm failure (5)  I $G(RSTYPE)="O" D SST^IBCNEUT2(TQN,3),RSTA^IBCNEUT7(TQN)  ;  ; Update the TQ service date to the date in the response file  ; if they are different AND the Error Action <>  ; 'P' for 'Please submit original transaction'  ;  ; \*\*\* Temporary change to suppress update of service & freshness dates.  ; \*\*\* To reinstate, remove comment (;) from next line.  ;I TQN'="",$G(RSTYPE)="O" D  ;. S TQSRVDT=$P($G(^IBCN(365.1,TQN,0)),U,12)  ;. I RSRVDT'="",TQSRVDT'=RSRVDT,$G(ERACT)'="P" D SAVETQ^IBCNEUT2(TQN,RSRVDT)  ;. ; update freshness date by same delta  ;. D SAVFRSH^IBCNEUT5(TQN,+$$FMDIFF^XLFDT(RSRVDT,TQSRVDT,1))  ;  ; Check for error action  I $G(ERACT)'=""!($G(ERTXT)'="") S ERACT=$$ERRACT^IBCNEHLU(RIEN),ERCON=$P(ERACT,U,2),ERACT=$P(ERACT,U) D ERROR^IBCNEHL3(TQN,ERACT,ERCON,TRACE) G FILX  ;  ; Stop processing if identification response and not an active policy  S FILEIT=1  I $G(IIVSTAT)=6,TQN]"" D  . I TQDATA="" Q  . I IBQFL'="I" Q  . S FILEIT=0  I 'FILEIT G FILX  ;  ; If there is an associated buffer entry & one or both of the following  ; is true, stop filing (don't update buffer entry)  ; 1) buffer status is not 'Entered'  ; 2) the buffer entry is verified (\* symbol)  I BUFF'="",($P($G(^IBA(355.33,BUFF,0)),U,4)'="E")!($$SYMBOL^IBCNBLL(BUFF)="\*") G FILX  ;  ; Set buffer symbol based on value returned from EC  S SYMBOL=MAP(IIVSTAT)  ;  ; If there is an associated buffer entry, update the buffer entry w/  ; response data  I BUFF'="" D RP^IBCNEBF(RIEN,"",BUFF)  ;  ; If no associated buffer entry, create one & populate w/ response  ; data (routine call sets IBFDA)  I BUFF="" D RP^IBCNEBF(RIEN,1) S BUFF=+IBFDA,UP(365,RIEN\_",",.04)=BUFF  ;  ; Set eIV Processed Date to now  S UP(355.33,BUFF\_",",.15)=$$NOW^XLFDT()  D FILE^DIE("I","UP","ERROR") FILX ;  Q  ; AUTOUPD(RIEN) ;  ; Returns "1^file 2 ien^file 2.312 ien^2nd file 2.312 ien^Medicare flag^subscriber flag", if entry  ; in file 365 is eligible for auto-update, returns 0 otherwise.  ;  ; Medicare flag: 1 for Medicare, 0 otherwise  ; Subscriber flag: 1 if patient is the subscriber, 0 otherwise  ;  ; For non-Medicare response: 1st file 2.312 ien is set, 2nd file 2.312 ien is empty, pieces 5-7 are empty  ; For Medicare response: 1st file 2.312 ien contains ien for Medicare Part A, 2nd file 2.312 ien contains ien for Medicare Part B,  ; either one may be empty, but at least one of them is set if entry is eligible.  ;  ; RIEN - ien in file 365  ;  N APPIEN,GDATA,GIEN,GNAME,GNUM,GNUM1,GOK,IEN2,IEN312,IEN36,IDATA0,IDATA3,ISSUB,MWNRA,MWNRB,MWNRIEN,MWNRTYP  N ONEPOL,PIEN,RDATA0,RDATA1,RES,TQIEN,IDATA7,RDATA13,RDATA14   ; IB\*2.0\*497  S RES=0  I +$G(RIEN)'>0 Q RES  ; invalid ien for file 365  I $G(IIVSTAT)'=1 Q RES ; only auto-update 'active policy' responses  S RDATA0=$G(^IBCN(365,RIEN,0)),RDATA1=$G(^IBCN(365,RIEN,1))  S RDATA13=$G(^IBCN(365,RIEN,13)),RDATA14=$G(^IBCN(365,RIEN,14)) ; IB\*2.0\*497 longer fields for GROUP NAME, GROUP NUMBER, NAME OF INSURED, and SUBSCRIBER ID  S PIEN=$P(RDATA0,U,3) I +PIEN>0 S APPIEN=$$PYRAPP^IBCNEUT5("IIV",PIEN)  I +$G(APPIEN)'>0 Q RES  ; couldn't find eIV application entry  ; Check dictionary 365.1 MANUAL REQUEST DATE/TIME Flag, Quit if Set.  I $P(RDATA0,U,5)'="",$P($G(^IBCN(365.1,$P(RDATA0,U,5),3)),U,1)'="" Q RES  I $P(^IBE(365.12,PIEN,1,APPIEN,0),U,7)=0 Q RES  ; auto-accept is OFF  S IEN2=$P(RDATA0,U,2) I +IEN2'>0 Q RES  ; couldn't find patient  S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25),MWNRTYP=0,(MWNRA,MWNRB)=""  I PIEN=MWNRIEN S MWNRTYP=$$ISMCR^IBCNEHLU(RIEN)  S ONEPOL=$$ONEPOL^IBCNEHLU(PIEN,IEN2)  ; try to find a matching pat. insurance  S IEN36="" F  S IEN36=$O(^DIC(36,"AC",PIEN,IEN36)) Q:IEN36=""!(RES>0) D  .S IEN312="" F  S IEN312=$O(^DPT(IEN2,.312,"B",IEN36,IEN312)) Q:IEN312=""!(RES>0&('+MWNRTYP)) D  ..S IDATA0=$G(^DPT(IEN2,.312,IEN312,0)),IDATA3=$G(^DPT(IEN2,.312,IEN312,3))  ..S IDATA7=$G(^DPT(IEN2,.312,IEN312,7)) ; IB\*2.0\*497 (vd)  ..I $$EXPIRED^IBCNEDE2($P(IDATA0,U,4)) Q  ; Insurance policy has expired  ..S ISSUB=$$PATISSUB^IBCNEHLU(IDATA0)  ..; Patient is the subscriber  ..I ISSUB,'$$CHK1^IBCNEHL3 Q  ..; Patient is the dependent  ..I 'ISSUB,'$$CHK2^IBCNEHL3(MWNRTYP) Q  ..; check group number  ..S GNUM=$P(RDATA14,U,2),GIEN=+$P(IDATA0,U,18),GOK=1 ;IB\*2\*497 group number needs to be retrieved from new field  ..; check non-Medicare group number  ..I '+MWNRTYP D  Q:'GOK  ; Group number doesn't match  ...I 'ONEPOL D  ....I GIEN'>0 S GOK=0 Q  ....S GNUM1=$P($G(^IBA(355.3,GIEN,2)),U,2) ; IB\*2.0\*497 (vd)  ....I GNUM=""!(GNUM1="")!(GNUM'=GNUM1) S GOK=0  ....Q  ...I ONEPOL D  ....I GNUM'="",GIEN'="" S GNUM1=$P($G(^IBA(355.3,GIEN,2)),U,2) I GNUM1'="",GNUM'=GNUM1 S GOK=0 ; IB\*2.0\*497 (vd)  ....Q  ...Q  ..; check for Medicare part A/B  ..I +MWNRTYP D  Q:'GOK  ; Group number doesn't match  ...I GIEN'>0 S GOK=0 Q  ...S GDATA=$G(^IBA(355.3,GIEN,0))  ...I $P(GDATA,U,14)="A" D  ....I $P(MWNRTYP,U,2)="MA"!($P(MWNRTYP,U,2)="B") S MWNRA=IEN312 Q  ....S GOK=0  ....Q  ...I $P(GDATA,U,14)="B" D  ....I $P(MWNRTYP,U,2)="MB"!($P(MWNRTYP,U,2)="B") S MWNRB=IEN312 Q  ....S GOK=0  ....Q  ...Q  ..S RES=1\_U\_IEN2\_U\_$S(+MWNRTYP:MWNRA\_U\_MWNRB\_U\_1,1:IEN312\_U\_U\_0)  ..S $P(RES,U,6)=ISSUB  ..Q  .Q  Q RES  ; EBFILE(DFN,IEN312,RIEN,AFLG) ; file eligibility/benefit data from file 365 into file 2.312  ; DFN - file 2 ien  ; IEN312 - file 2.312 ien  ; RIEN - file 365 ien  ; AFLG - 1 if called from autoupdate, 0 if called from ins. buffer process entry  ; Returns "" on success, ERFLG on failure. Also called from ACCEPT^IBCNBAR for manual processing of ins. buffer entry.  ;  ;  N DA,DIK,DATA,DATA1,EBIENS,ERFLG,ERROR,GIEN,GSKIP,IENROOT,IENS,IENSTR,TYPE,TYPE1,Z,Z1,Z2  ; delete existing EB data  S DIK="^DPT("\_DFN\_",.312,"\_IEN312\_",6,",DA(2)=DFN,DA(1)=IEN312  S DA=0 F  S DA=$O(^DPT(DFN,.312,IEN312,6,DA)) Q:DA=""!(DA?1.A) D ^DIK  ;  ; /IB\*2.0\*506 Beginning  ; File the new Requested Service Date field (file #2.312,8.01) from the file #365,1.1 field,  ; if the Service Date is not present, then use the Eligibility Date which would be from the file #365,1.11 field  ; ALSO, file the new Requested Service Type field (file #2.312,8.02) from the file #365.02,.04 field.  N DIE,DR,NODE0,RSRVDT,RSTYPE,TQIEN  S TQIEN=$P($G(^IBCN(365,RIEN,0)),U,5),NODE0=$G(^IBCN(365.1,TQIEN,0)),RSTYPE=$P(NODE0,U,20)  S RSRVDT=$P($G(^IBCN(365,RIEN,1)),U,10) I RSRVDT="" S RSRVDT=$P(NODE0,U,12)  S DIE="^DPT("\_DFN\_",.312,",DA(1)=DFN,DA=IEN312,DR="8.01///"\_RSRVDT\_";8.02///"\_RSTYPE  D ^DIE  ; /IB\*2.0\*506 End  ;  ; file new EB data  S IENSTR=IEN312\_","\_DFN\_","  S GIEN=+$P($G(^DPT(DFN,.312,IEN312,0)),U,18)  S Z="" F  S Z=$O(^IBCN(365,RIEN,2,"B",Z)) Q:Z=""!$G(ERFLG) D  .S EBIENS=$O(^IBCN(365,RIEN,2,"B",Z,""))\_","\_RIEN\_","  .; if filing Medicare Part A/B data, make sure we only file the correct EB group  .S GSKIP=0 I GIEN>0 D  ..S TYPE=$$GET1^DIQ(365.02,EBIENS,.05)  ..S TYPE1=$P($G(^IBA(355.3,GIEN,0)),U,14)  ..I TYPE="MA",TYPE1="B" S GSKIP=1  ..I TYPE="MB",TYPE1="A" S GSKIP=1  ..Q  .I GSKIP Q  ; wrong Medicare Part A/B EB group - skip it  .D GETS^DIQ(365.02,EBIENS,"\*\*",,"DATA","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 Q  .; make sure we have data to file  .I '$D(DATA(365.02)) Q  .S IENS="+1,"\_IENSTR,Z1=$O(DATA(365.02,"")) M DATA1(2.322,IENS)=DATA(365.02,Z1)  .D UPDATE^DIE("E","DATA1","IENROOT","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 Q  .S IENS="+1,"\_IENROOT(1)\_","\_IENSTR K DATA1,IENROOT  .S Z2="" F  S Z2=$O(DATA(365.26,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3226,IENS)=DATA(365.26,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.27,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3227,IENS)=DATA(365.27,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.28,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3228,IENS)=DATA(365.28,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.29,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3229,IENS)=DATA(365.29,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.291,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.32291,IENS)=DATA(365.291,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.292,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.32292,IENS)=DATA(365.292,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .K DATA  .Q  Q $G(ERFLG)  ; |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNEHL1 ;DAOU/ALA - HL7 Process Incoming RPI Messages ;26-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*300,345,416,444,438,497,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process incoming IIV response messages.  ; This includes updating the record in the IIV Response File,  ; updating the Buffer record (if there is one and creating a new  ; one if there isn't) with the appropriate Buffer Symbol and data  ;  ; Variables  ; SEG = HL7 Segment Name  ; MSGID = Original Message Control ID  ; ACK = Acknowledgment (AA=Accepted, AE=Error)  ; ERTXT = Error Message Text  ; ERFLG = Error quit flag  ; ERACT = Error Action  ; ERCON = Error Condition  ; RIEN = Response Record IEN  ; IIVSTAT = EC generated flag interpreting status of response  ; 1 = + (auto-update requirement)  ; 6 = -  ; V = #  ; MAP = Array that maps EC's IIV status flag to IIV STATUS TABLE (#365.15) IEN  ; EN ; Entry Point  N AUTO,EBDA,ERFLG,ERROR,G2OFLG,HCT,HLCMP,HLREP,HLSCMP,IIVSTAT,IRIEN,MAP,MGRP,RIEN,RSUPDT,SEG,SUBID,TRACE,UP,ACK  S (ERFLG,G2OFLG)=0,MGRP=$$MGRP^IBCNEUT5(),HCT=1,SUBID="",IIVSTAT=""  ;  S HLCMP=$E(HL("ECH")) ; HL7 component separator  S HLSCMP=$E(HL("ECH"),4) ; HL7 subcomponent separator  S HLREP=$E(HL("ECH"),2) ; HL7 repetition separator  ; Create map from EC to VistA  S MAP(1)=8,MAP(6)=9,MAP("V")=21  ;  ; Loop through the message and find each segment for processing  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:ERFLG  .D SPAR^IBCNEHLU  .S SEG=$G(IBSEG(1))  .; check if we are inside G2O group of segments  .I SEG="ZTY" S G2OFLG=1  .I G2OFLG,SEG'="ZTY",SEG'="CTD" S G2OFLG=0  .; If we are outside of Z\_Benefit\_group, kill EB multiple ien  .; I +$G(EBDA),".MSH.MSA.PRD.PID.GT1.IN1.IN3."[("."\_SEG\_".")!('G2OFLG&(SEG="CTD")) K EBDA  .;  .Q:SEG="PRD"  ; IB\*2\*497 PRD segment is not processed  .;  .I SEG="MSA" D MSA^IBCNEHL2(.ERACT,.ERCON,.ERROR,.ERTXT,.IBSEG,MGRP,.RIEN,.TRACE) Q  .;  .; Contact Segment  .I SEG="CTD",'G2OFLG D CTD^IBCNEHL2(.ERROR,.IBSEG,RIEN) Q  .;  .; Patient Segment  .I SEG="PID" D PID^IBCNEHL2(.ERFLG,.ERROR,.IBSEG,RIEN) Q  .;  .; Guarantor Segment  .I SEG="GT1" D GT1^IBCNEHL2(.ERROR,.IBSEG,RIEN,.SUBID) Q  .;  .; Insurance Segment  .I SEG="IN1" D IN1^IBCNEHL2(.ERROR,.IBSEG,RIEN,SUBID) Q  .;  .; Addt'l Insurance Segment  .;I SEG="IN2" ; for future expansion, add IN2 tag to IBCNEHL2  .;  .; Addt'l Insurance - Cert Segment  .I SEG="IN3" D IN3^IBCNEHL2(.ERROR,.IBSEG,RIEN) Q   .;  .; IB\*2\*497 GROUP LEVEL REFERENCE ID segment (x12 loops 2100C and 2100D)  . I SEG="ZRF",'$D(EBDA) D GZRF^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  .;  .; Eligibility/Benefit Segment  .I SEG="ZEB" D ZEB^IBCNEHL2(.EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Healthcare Delivery Segment  .I SEG="ZHS" D ZHS^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit level Reference ID Segment (X12 loops 2110C and 2110D)  .I SEG="ZRF",+$G(EBDA) D ZRF^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  ;IB\*2\*497 add check to make sure z benefit group  .;  .; Subscriber Date Segment  .I SEG="ZSD" D ZSD^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Subscriber Additional Info Segment  .I SEG="ZII" D ZII^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Segment  .I SEG="ZTY" D ZTY^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Contact Segment  .I SEG="CTD",G2OFLG D G2OCTD^IBCNEHL4(EBDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Benefit Related Entity Notes Segment  .I SEG="NTE",+$G(EBDA) D EBNTE^IBCNEHL2(EBDA,.IBSEG,RIEN) Q  .;  .; Reject Reasons Segment  .I SEG="ERR" K ERDA D ERR^IBCNEHL4(.ERDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Notes Segment  .I SEG="NTE",'$D(EBDA),+$G(ERDA) D NTE^IBCNEHL4(ERDA,.ERROR,.IBSEG,RIEN) Q  .;  .; Subscriber date segment (subscriber level)  .I SEG="ZTP" D ZTP^IBCNEHL4(.ERROR,.IBSEG,RIEN) Q  . ; ib\*2\*497 - add processing for ROL, DG1, and ZMP segments  . ; Provider Code segment   . I SEG="ROL" D ROL^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  . ;  . ; Health Care Diagnosis Code segment  . I SEG="DG1" D DG1^IBCNEHL5(.ERROR,.IBSEG,RIEN) Q  .;  .; Military Personnel Information segment  . I SEG="ZMP" D ZMP^IBCNEHL5(.ERROR,.IBSEG,RIEN)  ;  S AUTO=$$AUTOUPD(RIEN)  I $G(ACK)'="AE",$G(ERACT)="",$G(ERTXT)="",'$D(ERROR),+AUTO D  Q  .D:$P(AUTO,U,3)'="" AUTOFIL($P(AUTO,U,2),$P(AUTO,U,3),$P(AUTO,U,6))  .D:$P(AUTO,U,4)'="" AUTOFIL($P(AUTO,U,2),$P(AUTO,U,4),$P(AUTO,U,6))  .Q  D FIL  Q  ;  ; ================================================================= AUTOFIL(DFN,IEN312,ISSUB) ; Finish processing the response message - file directly into patient insurance  ; ; 2.6.4.9 - Added FLD  N BUFF,DATA,ERROR,FLD,IENS,PREL,RDATA0,RDATA1,RDATA5,RDATA13,RSTYPE,TQN,TSTAMP,MIL,OKAY   ; IB\*2.0\*497 (vd) ;  Q:$G(RIEN)=""  S TSTAMP=$$NOW^XLFDT(),IENS=IEN312\_","\_DFN\_","  S RDATA0=$G(^IBCN(365,RIEN,0)),RDATA1=$G(^IBCN(365,RIEN,1)),RDATA5=$G(^IBCN(365,RIEN,5))  S RDATA13=$G(^IBCN(365,RIEN,13)) ; IB\*2.0\*497 (vd)  S TQN=$P(RDATA0,U,5),RSTYPE=$P(RDATA0,U,10)  **; Modified for IB\*2.0\*549 – Req # 2.6.4.9**  I ISSUB,DATE(2.312,IENS,7.01)=”” S DATA(2.312,IENS,7.01)=$P(RDATA13,U) ; name - IB\*2.0\*497 (vd)  I DATA(2.312,IENS,3.01)=”” S DATA(2.312,IENS,3.01)=$P(RDATA1,U,2) ; dob  I DATA(2.312,IENS,3.01)=”” S DATA(2.312,IENS,3.05)=$P(RDATA1,U,3) ; ssn  I ISSUB,DATA(2.312,IENS,6)=”” S DATA(2.312,IENS,6)=$P(RDATA1,U,8) ; whose insurance  ; pt. relationship (365,8.01) IB\*2\*497 code from 365,8.01 needs evaluation and possible conversion  S PREL=$$GET1^DIQ(365,RIEN,8.01) I ISSUB,DATA(2.312,IENS,4.03)=””,PREL'="" S DATA(2.312,IENS,4.03)=$$PREL^IBCNEHLU(2.312,4.03,PREL)   S DATA(2.312,IENS,1.03)=TSTAMP ; date last verified  S DATA(2.312,IENS,1.04)="" ; last verified by  S DATA(2.312,IENS,1.05)=TSTAMP ; date last edited  S DATA(2.312,IENS,1.06)="" ; last edited by  S DATA(2.312,IENS,1.09)=5 ; source of info = eIV  **;  ;subscriber address  ; 2.6.4.9 - First check to see if ANY of the subscriber address fields have a  ; value and only update if NONE of them do  S OKAY=1  F FLD=3.06,3.07,3.08,3.09,3.1,3.13,3.14 D  Q:'OKAY  . S:$$DIQ^GET1(2.312,IENS,FLD,"I")'="" OKAY=0  I OKAY D   . S DATA(2.312,IENS,3.06)=$P(RDATA5,U) ; street line 1  . S DATA(2.312,IENS,3.07)=$P(RDATA5,U,2) ; street line 2  . S DATA(2.312,IENS,3.08)=$P(RDATA5,U,3) ; city  . S DATA(2.312,IENS,3.09)=$P(RDATA5,U,4) ; state  . S DATA(2.312,IENS,3.1)=$P(RDATA5,U,5) ; zip  . S DATA(2.312,IENS,3.13)=$P(RDATA5,U,6) ; country  . S DATA(2.312,IENS,3.14)=$P(RDATA5,U,7) ; country subdivision;**  L +^DPT(DFN,.312,IEN312):15 I '$T D LCKERR^IBCNEHL3 D FIL Q  D FILE^DIE("ET","DATA","ERROR") I $D(ERROR) D WARN^IBCNEHL3 K ERROR D FIL G AUTOFILX  ;  ; set eIV auto-update field separately because of the trigger on field 1.05  K DATA S DATA(2.312,IENS,4.04)="YES" D FILE^DIE("ET","DATA","ERROR") I $D(ERROR) D WARN^IBCNEHL3 G AUTOFILX  S ERFLG=$$GRPFILE(DFN,IEN312,RIEN,1) I $G(ERFLG) G AUTOFILX  ;IB\*2\*497 file data at 2.312, 9, 10 and 11 subfiles; if error is produced update buffer entry and then quit processing  ; file new EB data  S ERFLG=$$EBFILE(DFN,IEN312,RIEN,1)  ; bail out if something went wrong during filing of EB data  I $G(ERFLG) G AUTOFILX  ; update insurance record ien in transmission queue  D UPDIREC^IBCNEHL3(RIEN,IEN312)  ; For an original response, set the Transmission Queue Status to 'Response Received' &  ; update remaining retries to comm failure (5)  I $G(RSTYPE)="O" D SST^IBCNEUT2(TQN,3),RSTA^IBCNEUT7(TQN)  ; update buffer file entry so only stub remains and status is changed  S BUFF=+$P($G(^IBCN(365,RIEN,0)),U,4)  I BUFF D  .D STATUS^IBCNBEE(BUFF,"A",0,0,0) ; update buffer entry's status to accepted  .D DELDATA^IBCNBED(BUFF) ; delete buffer's insurance/patient data  .Q AUTOFILX ;  L -^DPT(DFN,.312,IEN312)  Q  ; GRPFILE(DFN,IEN312,RIEN,AFLG) ; ib\*2\*497 file data at node 12 and at subfiles 2.312,9, 10 and 11  ; DFN - file 2 ien  ; IEN312 - file 2.312 ien  ; RIEN = file 365 ien  ; AFLG - 1 if called from autoupdate, 0 if called from ins. buffer process entry  ; output - returns 0 or 1  ; 0 - entry update received an error when attempting to file  ; 1 - successful update  N DA,Z,Z2,DATA12,IENS,IENS365,IENS312,REF,PROV,DIAG,REF3129,PROV332,DIAG3121,NODE,ERROR,ERFLG  ; retrieve external values of data located at node 12 of 365  S IENS=IEN312\_","\_DFN\_","  D GETS^DIQ(365,RIEN,"12.01:12.07",,"MIL")  M DATA12(2.312,IENS)=MIL(365,RIEN\_",")  D FILE^DIE("ET","DATA12","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; remove existing sub-file entries at nodes 9, 10, and 11 before update of new data  F NODE="9","10","11" D  . S DIK="^DPT("\_DFN\_",.312,"\_IEN312\_","\_NODE\_",",DA(2)=DFN,DA(1)=IEN312  . S DA=0 F  S DA=$O(^DPT(DFN,.312,IEN312,NODE,DA)) Q:DA=""!(DA?1.A) D ^DIK  S IENS312="+1,"\_IEN312\_","\_DFN\_","  ; update node 9 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,9,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,9,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.09,IENS365,"\*",,"REF")  S Z2="" F  S Z2=$O(REF(365.09,Z2)) Q:Z2=""  M REF3129(2.3129,IENS312)=REF(365.09,Z2) D UPDATE^DIE("E","REF3129",,"ERROR") K REF3129 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; update node 10 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,10,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,10,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.04,IENS365,"\*",,"PROV")  S Z2="" F  S Z2=$O(PROV(365.04,Z2)) Q:Z2=""  M PROV332(2.332,IENS312)=PROV(365.04,Z2) D UPDATE^DIE("E","PROV332",,"ERROR") K PROV332 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR  ; update node 11 data  S Z="" F  S Z=$O(^IBCN(365,RIEN,11,"B",Z)) Q:'Z  D  . S IENS365=$O(^IBCN(365,RIEN,11,"B",Z,""))\_","\_RIEN\_","  . D GETS^DIQ(365.01,IENS365,"\*",,"DIAG")  S Z2="" F  S Z2=$O(DIAG(365.01,Z2)) Q:Z2=""  M DIAG3121(2.31211,IENS312)=DIAG(365.01,Z2) D UPDATE^DIE("E","DIAG3121",,"ERROR") K DIAG3121 I $D(ERROR) D:AFLG WARN^IBCNEHL3 K ERROR GRPFILEX ;  Q $G(ERFLG)  ; FIL ; Finish processing the response message - file into insurance buffer  ;  ; Input Variables  ; ERACT, ERFLG, ERROR, IIVSTAT, MAP, RIEN, TRACE  ;  ; If no record IEN, quit  I $G(RIEN)="" Q  ;  N BUFF,DFN,FILEIT,IBFDA,IBIEN,IBQFL,RDAT0,RSRVDT,RSTYPE,SYMBOL,TQDATA,TQN,TQSRVDT  ; Initialize variables from the Response File  S RDAT0=$G(^IBCN(365,RIEN,0)),TQN=$P(RDAT0,U,5)  S TQDATA=$G(^IBCN(365.1,TQN,0))  S IBQFL=$P(TQDATA,U,11)  S DFN=$P(RDAT0,U,2),BUFF=$P(RDAT0,U,4)  S IBIEN=$P(TQDATA,U,5),RSTYPE=$P(RDAT0,U,10)  S RSRVDT=$P($G(^IBCN(365,RIEN,1)),U,10)  ;  ; If an unknown error action or an error filing the response message,  ; send a warning email message  ; Note - A call to UEACT will always set ERFLAG=1  ;  ; IB\*2.0\*506 Removed the following line of code to Treat all AAA Action Codes  ; as though the Payer/FSC Responded.  ;I ",W,X,R,P,C,N,Y,S,"'[(","\_$G(ERACT)\_",")&($G(ERACT)'="")!$D(ERROR) D UEACT^IBCNEHL3  ;  ; If an error occurred, processing complete  I $G(ERFLG)=1 Q  ;  ; For an original response, set the Transmission Queue Status to 'Response Received' &  ; update remaining retries to comm failure (5)  I $G(RSTYPE)="O" D SST^IBCNEUT2(TQN,3),RSTA^IBCNEUT7(TQN)  ;  ; Update the TQ service date to the date in the response file  ; if they are different AND the Error Action <>  ; 'P' for 'Please submit original transaction'  ;  ; \*\*\* Temporary change to suppress update of service & freshness dates.  ; \*\*\* To reinstate, remove comment (;) from next line.  ;I TQN'="",$G(RSTYPE)="O" D  ;. S TQSRVDT=$P($G(^IBCN(365.1,TQN,0)),U,12)  ;. I RSRVDT'="",TQSRVDT'=RSRVDT,$G(ERACT)'="P" D SAVETQ^IBCNEUT2(TQN,RSRVDT)  ;. ; update freshness date by same delta  ;. D SAVFRSH^IBCNEUT5(TQN,+$$FMDIFF^XLFDT(RSRVDT,TQSRVDT,1))  ;  ; Check for error action  I $G(ERACT)'=""!($G(ERTXT)'="") S ERACT=$$ERRACT^IBCNEHLU(RIEN),ERCON=$P(ERACT,U,2),ERACT=$P(ERACT,U) D ERROR^IBCNEHL3(TQN,ERACT,ERCON,TRACE) G FILX  ;  ; Stop processing if identification response and not an active policy  S FILEIT=1  I $G(IIVSTAT)=6,TQN]"" D  . I TQDATA="" Q  . I IBQFL'="I" Q  . S FILEIT=0  I 'FILEIT G FILX  ;  ; If there is an associated buffer entry & one or both of the following  ; is true, stop filing (don't update buffer entry)  ; 1) buffer status is not 'Entered'  ; 2) the buffer entry is verified (\* symbol)  I BUFF'="",($P($G(^IBA(355.33,BUFF,0)),U,4)'="E")!($$SYMBOL^IBCNBLL(BUFF)="\*") G FILX  ;  ; Set buffer symbol based on value returned from EC  S SYMBOL=MAP(IIVSTAT)  ;  ; If there is an associated buffer entry, update the buffer entry w/  ; response data  I BUFF'="" D RP^IBCNEBF(RIEN,"",BUFF)  ;  ; If no associated buffer entry, create one & populate w/ response  ; data (routine call sets IBFDA)  I BUFF="" D RP^IBCNEBF(RIEN,1) S BUFF=+IBFDA,UP(365,RIEN\_",",.04)=BUFF  ;  ; Set eIV Processed Date to now  S UP(355.33,BUFF\_",",.15)=$$NOW^XLFDT()  D FILE^DIE("I","UP","ERROR") FILX ;  Q  ; AUTOUPD(RIEN) ;  ; Returns "1^file 2 ien^file 2.312 ien^2nd file 2.312 ien^Medicare flag^subscriber flag", if entry  ; in file 365 is eligible for auto-update, returns 0 otherwise.  ;  ; Medicare flag: 1 for Medicare, 0 otherwise  ; Subscriber flag: 1 if patient is the subscriber, 0 otherwise  ;  ; For non-Medicare response: 1st file 2.312 ien is set, 2nd file 2.312 ien is empty, pieces 5-7 are empty  ; For Medicare response: 1st file 2.312 ien contains ien for Medicare Part A, 2nd file 2.312 ien contains ien for Medicare Part B,  ; either one may be empty, but at least one of them is set if entry is eligible.  ;  ; RIEN - ien in file 365  ;  N APPIEN,GDATA,GIEN,GNAME,GNUM,GNUM1,GOK,IEN2,IEN312,IEN36,IDATA0,IDATA3,ISSUB,MWNRA,MWNRB,MWNRIEN,MWNRTYP  N ONEPOL,PIEN,RDATA0,RDATA1,RES,TQIEN,IDATA7,RDATA13,RDATA14   ; IB\*2.0\*497  S RES=0  I +$G(RIEN)'>0 Q RES  ; invalid ien for file 365  I $G(IIVSTAT)'=1 Q RES ; only auto-update 'active policy' responses  S RDATA0=$G(^IBCN(365,RIEN,0)),RDATA1=$G(^IBCN(365,RIEN,1))  S RDATA13=$G(^IBCN(365,RIEN,13)),RDATA14=$G(^IBCN(365,RIEN,14)) ; IB\*2.0\*497 longer fields for GROUP NAME, GROUP NUMBER, NAME OF INSURED, and SUBSCRIBER ID  S PIEN=$P(RDATA0,U,3) I +PIEN>0 S APPIEN=$$PYRAPP^IBCNEUT5("IIV",PIEN)  I +$G(APPIEN)'>0 Q RES  ; couldn't find eIV application entry  ; Check dictionary 365.1 MANUAL REQUEST DATE/TIME Flag, Quit if Set.  I $P(RDATA0,U,5)'="",$P($G(^IBCN(365.1,$P(RDATA0,U,5),3)),U,1)'="" Q RES  I $P(^IBE(365.12,PIEN,1,APPIEN,0),U,7)=0 Q RES  ; auto-accept is OFF  S IEN2=$P(RDATA0,U,2) I +IEN2'>0 Q RES  ; couldn't find patient  S MWNRIEN=$P($G(^IBE(350.9,1,51)),U,25),MWNRTYP=0,(MWNRA,MWNRB)=""  I PIEN=MWNRIEN S MWNRTYP=$$ISMCR^IBCNEHLU(RIEN)  S ONEPOL=$$ONEPOL^IBCNEHLU(PIEN,IEN2)  ; try to find a matching pat. insurance  S IEN36="" F  S IEN36=$O(^DIC(36,"AC",PIEN,IEN36)) Q:IEN36=""!(RES>0) D  .S IEN312="" F  S IEN312=$O(^DPT(IEN2,.312,"B",IEN36,IEN312)) Q:IEN312=""!(RES>0&('+MWNRTYP)) D  ..S IDATA0=$G(^DPT(IEN2,.312,IEN312,0)),IDATA3=$G(^DPT(IEN2,.312,IEN312,3))  ..S IDATA7=$G(^DPT(IEN2,.312,IEN312,7)) ; IB\*2.0\*497 (vd)  ..I $$EXPIRED^IBCNEDE2($P(IDATA0,U,4)) Q  ; Insurance policy has expired  ..S ISSUB=$$PATISSUB^IBCNEHLU(IDATA0)  ..; Patient is the subscriber  ..I ISSUB,'$$CHK1^IBCNEHL3 Q  ..; Patient is the dependent  ..I 'ISSUB,'$$CHK2^IBCNEHL3(MWNRTYP) Q  ..; check group number  ..S GNUM=$P(RDATA14,U,2),GIEN=+$P(IDATA0,U,18),GOK=1 ;IB\*2\*497 group number needs to be retrieved from new field  ..; check non-Medicare group number  ..I '+MWNRTYP D  Q:'GOK  ; Group number doesn't match  ...I 'ONEPOL D  ....I GIEN'>0 S GOK=0 Q  ....S GNUM1=$P($G(^IBA(355.3,GIEN,2)),U,2) ; IB\*2.0\*497 (vd)  ....I GNUM=""!(GNUM1="")!(GNUM'=GNUM1) S GOK=0  ....Q  ...I ONEPOL D  ....I GNUM'="",GIEN'="" S GNUM1=$P($G(^IBA(355.3,GIEN,2)),U,2) I GNUM1'="",GNUM'=GNUM1 S GOK=0 ; IB\*2.0\*497 (vd)  ....Q  ...Q  ..; check for Medicare part A/B   ..I +MWNRTYP D  Q:'GOK  ; Group number doesn't match  ...I GIEN'>0 S GOK=0 Q  ...S GDATA=$G(^IBA(355.3,GIEN,0)) **. . ; IB\*2.0\*549 – Only want to update active Medicare plans. – Req # 2.6.4.8  . . I +$P(GDATA,U,11) S GOK=0 Q    ; 0=ACTIVE, 1=INACTIVE**   ...I $P(GDATA,U,14)="A" D  ....I $P(MWNRTYP,U,2)="MA"!($P(MWNRTYP,U,2)="B") S MWNRA=IEN312 Q  ....S GOK=0  ....Q  ...I $P(GDATA,U,14)="B" D  ....I $P(MWNRTYP,U,2)="MB"!($P(MWNRTYP,U,2)="B") S MWNRB=IEN312 Q  ....S GOK=0  ....Q  ...Q  ..S RES=1\_U\_IEN2\_U\_$S(+MWNRTYP:MWNRA\_U\_MWNRB\_U\_1,1:IEN312\_U\_U\_0)  ..S $P(RES,U,6)=ISSUB  ..Q  .Q  Q RES  ; EBFILE(DFN,IEN312,RIEN,AFLG) ; file eligibility/benefit data from file 365 into file 2.312  ; DFN - file 2 ien  ; IEN312 - file 2.312 ien  ; RIEN - file 365 ien  ; AFLG - 1 if called from autoupdate, 0 if called from ins. buffer process entry  ; Returns "" on success, ERFLG on failure. Also called from ACCEPT^IBCNBAR for manual processing of ins. buffer entry.  ;  ;  N DA,DIK,DATA,DATA1,EBIENS,ERFLG,ERROR,GIEN,GSKIP,IENROOT,IENS,IENSTR,TYPE,TYPE1,Z,Z1,Z2  ; delete existing EB data  S DIK="^DPT("\_DFN\_",.312,"\_IEN312\_",6,",DA(2)=DFN,DA(1)=IEN312  S DA=0 F  S DA=$O(^DPT(DFN,.312,IEN312,6,DA)) Q:DA=""!(DA?1.A) D ^DIK  ;  ; /IB\*2.0\*506 Beginning  ; File the new Requested Service Date field (file #2.312,8.01) from the file #365,1.1 field,  ; if the Service Date is not present, then use the Eligibility Date which would be from the file #365,1.11 field  ; ALSO, file the new Requested Service Type field (file #2.312,8.02) from the file #365.02,.04 field. **;/IB\*2.0\*549 – file the new Trace Number field (file #2.312,8.03) from the file #365,.09 field. – Req #2.6.4.4**  N DIE,DR,NODE0,RSRVDT,RSTYPE,TQIEN**,TRNUM**  S TQIEN=$P($G(^IBCN(365,RIEN,0)),U,5)**,TRNUM=$P($G(^IBCN(365,RIEN,0)),U,9),NODE0=$G(^IBCN(365.1,TQIEN,0)),RSTYPE=$P(NODE0,U,20)**  S RSRVDT=$P($G(^IBCN(365,RIEN,1)),U,10) I RSRVDT="" S RSRVDT=$P(NODE0,U,12)  S DIE="^DPT("\_DFN\_",.312,",DA(1)=DFN,DA=IEN312,DR="8.01///"\_RSRVDT\_";8.02///"\_RSTYPE\_";8.03///"**\_TRNUM**   N DIE,DR,NODE0,RSRVDT,RSTYPE,TQIEN  S TQIEN=$P($G(^IBCN(365,RIEN,0)),U,5),NODE0=$G(^IBCN(365.1,TQIEN,0)),RSTYPE=$P(NODE0,U,20)  S RSRVDT=$P($G(^IBCN(365,RIEN,1)),U,10) I RSRVDT="" S RSRVDT=$P(NODE0,U,12)  S DIE="^DPT("\_DFN\_",.312,",DA(1)=DFN,DA=IEN312,DR="8.01///"\_RSRVDT\_";8.02///"\_RSTYPE  D ^DIE  ; /IB\*2.0\*506 End  ;  ; file new EB data  S IENSTR=IEN312\_","\_DFN\_","  S GIEN=+$P($G(^DPT(DFN,.312,IEN312,0)),U,18)  S Z="" F  S Z=$O(^IBCN(365,RIEN,2,"B",Z)) Q:Z=""!$G(ERFLG) D  .S EBIENS=$O(^IBCN(365,RIEN,2,"B",Z,""))\_","\_RIEN\_","  .; if filing Medicare Part A/B data, make sure we only file the correct EB group  .S GSKIP=0 I GIEN>0 D  ..S TYPE=$$GET1^DIQ(365.02,EBIENS,.05)  ..S TYPE1=$P($G(^IBA(355.3,GIEN,0)),U,14)  ..I TYPE="MA",TYPE1="B" S GSKIP=1  ..I TYPE="MB",TYPE1="A" S GSKIP=1  ..Q  .I GSKIP Q  ; wrong Medicare Part A/B EB group - skip it  .D GETS^DIQ(365.02,EBIENS,"\*\*",,"DATA","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 Q  .; make sure we have data to file  .I '$D(DATA(365.02)) Q  .S IENS="+1,"\_IENSTR,Z1=$O(DATA(365.02,"")) M DATA1(2.322,IENS)=DATA(365.02,Z1)  .D UPDATE^DIE("E","DATA1","IENROOT","ERROR") I $D(ERROR) D:AFLG WARN^IBCNEHL3 Q  .S IENS="+1,"\_IENROOT(1)\_","\_IENSTR K DATA1,IENROOT  .S Z2="" F  S Z2=$O(DATA(365.26,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3226,IENS)=DATA(365.26,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.27,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3227,IENS)=DATA(365.27,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.28,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3228,IENS)=DATA(365.28,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.29,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.3229,IENS)=DATA(365.29,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.291,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.32291,IENS)=DATA(365.291,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .S Z2="" F  S Z2=$O(DATA(365.292,Z2)) Q:Z2=""!$G(ERFLG) D  ..M DATA1(2.32292,IENS)=DATA(365.292,Z2) D UPDATE^DIE("E","DATA1",,"ERROR") K DATA1 I $D(ERROR) D:AFLG WARN^IBCNEHL3  ..Q  .K DATA  .Q  Q $G(ERFLG); |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNBLL** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.4.7 | | | | | | | | |
| **Related Options** | Process Insurance Buffer [IBCN INSURANCE BUFFER PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBAR  IBCNBLA  IBCNBLA1  IBCNBLE1  IBCNEDE1  IBCNEDST  IBCNEHL1  IBCNERP8  IBCNEUT4 | | | | | LST^DGMTU  INSERROR^IBCNEUT3  INSURED^IBCNS1  ALL^IBCNS1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNB LIST COMPLETE VIEW, IBCNB LIST MEDICARE VIEW | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .  ; BLDLN(IBBUFDA,IBCNT,DFLG) ; build line to display on List screen for one Buffer entry  N DFN,IB0,IB20,IB60,IBLINE,IBY,VAIN,VADM,VA,VAERR,X,Y,IBMTS S IBLINE="",IBBUFDA=+$G(IBBUFDA)  S IB0=$G(^IBA(355.33,IBBUFDA,0)),IB20=$G(^IBA(355.33,IBBUFDA,20)),IB60=$G(^IBA(355.33,IBBUFDA,60))  S DFN=+IB60 I +DFN D DEM^VADPT,INP^VADPT  ;  I 'IBKEYS,'$$ACTIVE(DFN) G BLDLNQ  ;IB\*2\*506/taz Only allow active insurance for users not holding IB INSURANCE EDIT or IB GROUP/PLAN EDIT keys  ;  S IBY=$G(IBCNT),IBLINE=$$SETSTR^VALM1(IBY,"",1,4)  ;  ; ESG - 6/6/02 - SDD 5.1.8  ; pull the symbol from the symbol function  ;  S IBY=$$SYMBOL(IBBUFDA)  S IBY=IBY\_$P($G(^DPT(+DFN,0)),U,1),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,5,20)  S IBLINE=$$SETSTR^VALM1(DFLG,IBLINE,25,1)  S IBY=$G(VA("BID")),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,27,4)  S IBY=$P(IB20,U,1),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,32,17)  S IBY=$P(IB60,U,4),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,50,13)  S IBY=$$GET1^DIQ(355.12,$P(IB0,U,3),.03),IBLINE=$$SETSTR^VALM1($$SRCCNV(IBY),IBLINE,64,1)  S IBY=$$DATE(+IB0),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,66,8)  S IBY="" D  S IBLINE=$$SETSTR^VALM1(IBY,IBLINE,76,5)  . S IBY=IBY\_$S(+$$INSURED^IBCNS1(DFN,DT):"i",1:" ")  . S IBY=IBY\_$S(+$G(VAIN(1)):"I",1:" ")  . S IBY=IBY\_$S(+$G(VADM(6)):"E",1:" ")  . S IBMTS=$P($$LST^DGMTU(DFN),U,4)  . S IBY=IBY\_$S(IBMTS="C":"Y",IBMTS="G":"Y",1:" ")  . S IBY=IBY\_$S(+$$HOLD(DFN):"H",1:" ") BLDLNQ ; IB\*2\*506/taz Tag added  Q IBLINE  ; |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  .  ; BLDLN(IBBUFDA,IBCNT,DFLG) ; build line to display on List screen for one Buffer entry  **; 2.6.4.7 All changes in this method**  N DFN,IB0,IB20,**IB40**,IB60,IBLINE,IBY,**PLNTYP**,VAIN,VADM,VA,VAERR,X,Y,IBMTS  S IBLINE="",IBBUFDA=+$G(IBBUFDA)  S IB0=$G(^IBA(355.33,IBBUFDA,0)),IB20=$G(^IBA(355.33,IBBUFDA,20)),IB60=$G(^IBA(355.33,IBBUFDA,60))  **S IB40=$G(^IBA(355.33,IBBUFDA,40)),PLNTYP=$P(+$G(^IBE(355.1,+$P(IB40,U,9),0)),U,3)**  S DFN=+IB60 I +DFN D DEM^VADPT,INP^VADPT  ;  **; Replace the following line of code:  ;I 'IBKEYS,'$$ACTIVE(DFN) G BLDLNQ  ;IB\*2\*506/taz Only allow active insurance for users not holding IB INSURANCE EDIT or IB GROUP/PLAN EDIT keys**  **; with code that will determine if the list item is Medicare (PLNTYP=5) and if the list item is Medicare,**  **; then include it on the list even if the user doesn’t have the security keys or if the patient has or has not**  **; ACTIVE policies.  I 'IBKEYS,'$$ACTIVE(DFN),+PLNTYP’=5 G BLDLNQ  ;IB\*2\*549/vad – exclude item if it is not Medicare and the user doesn’t have the appropriate security keys and no matter if the Patient has or has not Active policies.**  ;  S IBY=$G(IBCNT),IBLINE=$$SETSTR^VALM1(IBY,"",1,4)  ;  ; ESG - 6/6/02 - SDD 5.1.8  ; pull the symbol from the symbol function  ;  S IBY=$$SYMBOL(IBBUFDA)  S IBY=IBY\_$P($G(^DPT(+DFN,0)),U,1),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,5,20)  S IBLINE=$$SETSTR^VALM1(DFLG,IBLINE,25,1)  S IBY=$G(VA("BID")),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,27,4)  S IBY=$P(IB20,U,1),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,32,17)  S IBY=$P(IB60,U,4),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,50,13)  S IBY=$$GET1^DIQ(355.12,$P(IB0,U,3),.03),IBLINE=$$SETSTR^VALM1($$SRCCNV(IBY),IBLINE,64,1)  S IBY=$$DATE(+IB0),IBLINE=$$SETSTR^VALM1(IBY,IBLINE,66,8)  S IBY="" D  S IBLINE=$$SETSTR^VALM1(IBY,IBLINE,76,5)  . S IBY=IBY\_$S(+$$INSURED^IBCNS1(DFN,DT):"i",1:" ")  . S IBY=IBY\_$S(+$G(VAIN(1)):"I",1:" ")  . S IBY=IBY\_$S(+$G(VADM(6)):"E",1:" ")  . S IBMTS=$P($$LST^DGMTU(DFN),U,4)  . S IBY=IBY\_$S(IBMTS="C":"Y",IBMTS="G":"Y",1:" ")  . S IBY=IBY\_$S(+$$HOLD(DFN):"H",1:" ") BLDLNQ ; IB\*2\*506/taz Tag added  Q IBLINE  ;  .  .  . |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBY549PO** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.2.2, 2.6.2.4, 2.6.4.10, 2.6.8.1, 2.6.8.2, 2.6.5.2 | | | | | | | | |
| **Related Options** | NONE | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBY549PO ;ALB/VD - Post-Install for IB patch 549 ;09-APR-2015  ;;2.0;INTEGRATED BILLING;\*\*549\*\*;APR 09, 2015  ;;Per VA Directive 6402, this routine should not be modified.  ; EN ; Entry point  N IBXPD,XPDIDTOT  S XPDIDTOT=1  ;  ;  S IBXPD=1  D IIVEC(IBXPD,XPDIDTOT) ;2.6.4.10 Remove the IP Address if it exists in the IIV EC HL7 Logical Link.  ;  S IBXPD=2  D NEWCVTY(IBXPD,XPDIDTOT) ;2.6.8.1 Add a new Type of Coverage record in the ^IBE(355.2) file.  ;  S IBXPD=3  D NEWPLTY(IBXPD,XPDIDTOT) ;2.6.8.2 Add a new Type of Plan record in file 355.1.  ;  S IBXPD=4  D POLCYUPD(IBXPD,XPDIDTOT) ;2.6.2.2, 2.6.2.4 Update Policy expiration dates for deceased patients**  **S IBXPD=5 ; Req# 2.6.5.2  ; Flip the 'Policy Not Billable' field (#36,3.04) for every patient  D POLCYNB(IBXPD,XPDIDTOT)**  **;**  **; … There will most likely be more calls added to this routine.**  **;  Q  ;**  **IIVEC(IBXPD,XPDIDTOT) ;2.6.4.10 Remove the IP address if it exists in the IIV EC HL7 Logical Link.**  **D BMES^XPDUTL(" STEP "\_IBXPD\_" of "\_XPDIDTOT)  D MES^XPDUTL("-------------")  D MES^XPDUTL("Remove the IP Address in ‘IIV EC’ HL7 Logical Link…")  ;  S DIE="^HLCS(870,",DA=”IIV EC”,DR="400.01///@" D ^DIE**  **D DONE**  **Q  ;**  **NEWPLTY(IBXPD,XPDIDTOT) ;2.6.8.2 add a new code to the TYPE OF PLAN TABLE (#355.1) for VA SPECIAL CLASS  D BMES^XPDUTL(" STEP "\_IBXPD\_" of "\_XPDIDTOT)  D MES^XPDUTL("-------------")  D MES^XPDUTL("Add a new VA SPECIAL CLASS code to the TYPE OF PLAN TABLE")  N IBACTN,IBDATA,IBDESC,IBERR,IBIEN  I $D(^IBE(355.1,"D","VSC")) D BMES^XPDUTL("\*\*\* NEW 'VSC' CODE NOT ADDED TO TYPE OF PLAN TABLE...ALREADY EXISTS \*\*\*") G NEWPLTYX  ;  ;Set up WP Arrays  S IBDESC("WP",1)="Need to add description of what this type of plan represents…see other"  S IBDESC("WP",2)="types of coverage records for examples."  ;  S IBACTN("WP",1)="Need to add ‘Action to take:’ Description to better identify what this action"  S IBACTN("WP",2)="represents."  ;  ;Set up File Nodes  S IBDATA(.01)="VA SPECIAL CLASS"  S IBDATA(.02)=”VSC”  S IBDATA(.03)=<MAJOR CATEGORY> … MUST BE A NUMBER 1-7 FROM THE FOLLOWING:  ; '1' FOR MAJOR MEDICAL;   ; '2' FOR DENTAL;   ; '3' FOR HMO;   ; '4' FOR PPO;   ; '5' FOR MEDICARE;   ; '6' FOR MEDICAID;   ; '7' FOR CHAMPUS;   S IBDATA(1)=$NA(IBDESC("WP"))  S IBDATA(2)=$NA(IBACTN("WP"))  S IBIEN=$$ADD^IBDFDBS(355.1,,.IBDATA,.IBERR)  I IBERR D BMES^XPDUTL("\*\*\* ERROR ADDING ‘VSC' CODE TO THE TYPE OF PLAN TABLE (#355.1) \*\*\*") G NEWPLTYX  D DONE NEWPLTYX ;  Q  ; NEWCVTY(IBXPD,XPDIDTOT) ;2.6.8.1 add a new code to the TYPE OF COVERAGE TABLE (#355.2) for VA SPECIAL CLASS  D BMES^XPDUTL(" STEP "\_IBXPD\_" of "\_XPDIDTOT)  D MES^XPDUTL("-------------")  D MES^XPDUTL("Add a new VA SPECIAL CLASS code to the TYPE OF COVERAGE TABLE")  N IBACTN,IBDATA,IBDESC,IBERR,IBIEN  I $D(^IBE(355.2,"C","VSC")) D BMES^XPDUTL("\*\*\* NEW 'VSC' CODE NOT ADDED TO TYPE OF COVERAGE TABLE...ALREADY EXISTS \*\*\*") G NEWCVTYX  ;  ;Set up WP Arrays  S IBDESC("WP",1)="Need to add description of what this type of coverage represents…see other"  S IBDESC("WP",2)="types of coverage records for examples."  ;  S IBACTN("WP",1)="Need to add ‘Action to take:’ Description to better identify what this action"  S IBACTN("WP",2)="represents."  ;  ;Set up File Nodes  S IBDATA(.01)="VA SPECIAL CLASS"  S IBDATA(.02)=”VSC”  S IBDATA(1)=$NA(IBDESC("WP"))  S IBDATA(2)=$NA(IBACTN("WP"))  S IBIEN=$$ADD^IBDFDBS(355.2,,.IBDATA,.IBERR)  I IBERR D BMES^XPDUTL("\*\*\* ERROR ADDING ‘VSC' CODE TO THE TYPE OF COVERAGE TABLE (#355.2) \*\*\*") G NEWCVTYX  D DONE NEWCVTYX ;  Q  ; POLCYUPD ;2.6.2.2, 2.6.2.4 Update Policy Expiration Dates for deceased patients**  **; 1 - For every patient in the patient file check if the patient is deceased field (file 2, field .351)**  **; 2 - Quit if the patient is not deceased**  **; 3 - For every group plan of the deceased patient, check to see if the plan has an effective**  **; 4 - Quit if the effective date is null**  **; 5 – Quit if the plan expiration date < (Date of Death+1)**  **; 6 – Change the group plan expiration date to (Date of Death +1)**  **; 7 – Add this patient and group plan to a list of patient/plans being modified**  **; 8 – Email the list of patient/plans modified to xxx**  **Q**  **; POLCYNB(IBXPD,XPDIDTOT) ; Update Policy Not Billable field (3.04) for every  ; Patient. Changed the prompt from 'Policy Not Billable' to 'Is this Policy Billable'  ; which necessitates the flipping of all currently stored valued  N DA,DFN,DIE,XX  D BMES^XPDUTL(" STEP "\_IBXPD\_" of "\_XPDIDTOT)  D MES^XPDUTL("-------------")  D MES^XPDUTL("Update POLICY NOT BILLABLE (field 3.04) for all patients.")  I IBERR D BMES^XPDUTL("\*\*\* ERROR UPDATING POLICY NOT BILLABLE FIELD FOR ALL PATIENTS \*\*\*") G POLCYUPDX  D DONE  S DFN=0  F  D  Q:+DFN=0  . S DFN=$O(^DPT(DFN))  . Q:+DFN=0  . S XX=$$GET1^DIQ(2.312,DFN\_",",3.04,,"I")  . S XX=$S(XX="Y":"N",1:"Y")  . S DA=DFN  . S DR="3.04///"\_XX  . D ^DIE  Q  ; DONE ; Displays the 'Done' message and finishes the progress bar  ; Input: IBXPD - Post-Installation step being performed  D MES^XPDUTL(" Done.")  Q  ;** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNICB** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.6.1, 2.6.6.2 | | | | | | | | |
| **Related Options** | Process Insurance Buffer [IBCN INSURANCE BUFFER PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBAR  IBCNBMI | | | | | PROCESS^IBCNBAR  REJPROC^IBCNBAR  SUBS^IBCNSJ  DBU^IBCNSJ  DEL^IBCNSJ  IR^IBCNSJ21  BU^IBCNSJ21 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .  ;  ; ACCEPAPI(RESULT,IBBUFDA,DFN,IBINSDA,IBGRPDA,IBPOLDA,IBMVINS,IBMVGRP,IBMVPOL,IBNEWINS,IBNEWGRP,IBNEWPOL,IVMREPTR,IBELIG) ;  ;Provides API to be called by the Insurance Capture Buffer (ICB)   ;application to move buffer data in Insurance Files then cleanup  ;  ;The call to PROCESS^IBCNBAR and embedded Sub calls are updated to   ;provide data in the RESULT parameter and suppress user I/O when   ;function is called by ICB.  ;Input  ; IBBUFDA - INSURANCE BUFFER (#355.33) file internal entry number  ; (IEN) (Required)  ; DFN - PATIENT (#2) file IEN (Required)  ; IBINSDA - INSURANCE COMPANY (#36) File IEN if not adding new entry  ; (Optional)  ; IBGRPDA - GROUP INSURANCE PLAN (#355.3) File IEN if not adding new  ; entry (Optional)  ; IBPOLDA - INSURANCE TYPE (#2.312) sub-file of PATIENT (#2) IEN if  ; not adding new entry (Optional)  ; IBMVINS - Type for INSURANCE (Required)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBMVGRP - Type for GROUP (Optional)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBMVPOL - Type for POLICY (Optional)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBNEWINS - Add new INSURANCE COMPANY flag (non-zero means add)  ; IBNEWGRP - Add new GROUP INSURANCE PLAN flag (non-zero means add)  ; IBNEWPOL - Add new patient insurance policy (non-zero means add)  ; IVMREPTR - IVM REASONS FOR NOT UPLOADING (#301.91) IEN (Optional)  ;  ;OUTPUT  ; RESULT - Returned Parameter Array with IENS of new entries and/or   ; errors/warning.  ; RESULT(0) = -1^error message  ; RESULT(0) = 0 -Move worked  ; RESULT(0) = 0 ^ warning message ^ warning message ^   ; warning message ^ warning message  ; - Move worked but there may be zero to 4 warning messages  ; RESULT(1) = "IBINSDA^" IEN of new Insurance Company (#36) File  ; RESULT(1,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 36.  ; RESULT(2) = "IBGRPDA^" IEN of new GROUP INSURANCE PLAN (#355.3)  ; File  ; RESULT(2,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 355.3.  ; RESULT(3) = "IBPOLDA^" IEN of new INSURANCE TYPE (#2.312) sub-file  ; of PATIENT (#2) IEN  ; RESULT(3,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 2.312.  ; RESULT(4) Contains the results of the call to UPDPOL^IBCNICB which  ; is used to update a new group into an existing patient  ; policy entry when applicable.  ; RESULT(4) =-1^error message   ; RESULT(4) =0 ^ message that process was successful or not required  ;  N IBSUPRES,IBUFSTAT,IBX  I '$D(IBELIG) S IBELIG=0  ;Set IBSUPRES to suppress screen I/O within ACCEPT  S IBSUPRES=1,IBUFSTAT=$P($G(^IBA(355.33,$G(IBBUFDA),0)),U,4)  ;  S RESULT(0)="-1^INSURANCE BUFFER (#355.33) IEN required" Q:'$G(IBBUFDA)  S RESULT(0)="-1^INSURANCE BUFFER ENTRY PREVIOUSLY PROCESSED"  Q:"~A~R~"[("~"\_IBUFSTAT\_"~")  S RESULT(0)="-1^INSURANCE BUFFER ENTRY STATUS SHOULD BE ENTERED"  Q:IBUFSTAT'="E"  S RESULT(0)="-1^PATIENT (#2) IEN required" Q:'$G(DFN)  S IBINSDA=$G(IBINSDA),IBGRPDA=$G(IBGRPDA),IBPOLDA=$G(IBPOLDA)  S IBMVINS=$G(IBMVINS,2),IBMVGRP=$G(IBMVGRP,2),IBMVPOL=$G(IBMVPOL,2)  S IBNEWINS=$G(IBNEWINS),IBNEWGRP=$G(IBNEWGRP),IBNEWPOL=$G(IBNEWPOL)  ;  S RESULT(0)="-1^Passed INSURANCE COMPANY (#36) entry doesn't exist"  I +IBINSDA,$G(^DIC(36,IBINSDA,0))="" Q  S RESULT(0)="-1^Passed GROUP INSURANCE PLAN (#355.3) entry doesn't exist"  I +IBGRPDA,$G(^IBA(355.3,IBGRPDA,0))="" Q S RESULT(0)="-1^Passed Patient INSURANCE TYPE (#2.312) entry doesn't exist"  I +IBPOLDA,$G(^DPT(DFN,.312,IBPOLDA,0))="" Q  S RESULT(0)="-1^Passed GROUP INSURANCE PLAN (#355.3) entry points to different INSURANCE COMPANY (#36) entry"  I +IBGRPDA,+IBINSDA,+$G(^IBA(355.3,IBGRPDA,0))'=IBINSDA Q  S RESULT(0)="-1^Individual Policy Patient required to be Patient DFN when Group Insurance Plan is not Group Policy"  I +IBGRPDA S IBX=$G(^IBA(355.3,IBGRPDA,0)) I $P(IBX,U,2)=0,+$P(IBX,U,10),$P(IBX,U,10)'=DFN Q  ;  ;If existing GROUP INSURANCE PLAN (#355.3) entry is being changed from   ;a group plan to individual plan with other subscribers, send error   ;message and abort update  S RESULT(0)="-1^Can't change GROUP INSURANCE PLAN from Group Plan to Individual Plan when there are subscribers"  I +IBGRPDA,$P(IBX,U,2)=1,$P($G(^IBA(355.33,+$G(IBBUFDA),40)),U,1)'=1,$$SUBS^IBCNSJ(IBINSDA,IBGRPDA)>1 Q  ;  D PROCESS^IBCNBAR  Q  ;.  . |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  .  ;  ; ACCEPAPI(RESULT,IBBUFDA,DFN,IBINSDA,IBGRPDA,IBPOLDA,IBMVINS,IBMVGRP,IBMVPOL,IBNEWINS,IBNEWGRP,IBNEWPOL,IVMREPTR,IBELIG) ;  ;Provides API to be called by the Insurance Capture Buffer (ICB)   ;application to move buffer data in Insurance Files then cleanup  ;  ;The call to PROCESS^IBCNBAR and embedded Sub calls are updated to   ;provide data in the RESULT parameter and suppress user I/O when   ;function is called by ICB.  ;Input  ; IBBUFDA - INSURANCE BUFFER (#355.33) file internal entry number  ; (IEN) (Required)  ; DFN - PATIENT (#2) file IEN (Required)  ; IBINSDA - INSURANCE COMPANY (#36) File IEN if not adding new entry  ; (Optional)  ; IBGRPDA - GROUP INSURANCE PLAN (#355.3) File IEN if not adding new  ; entry (Optional)  ; IBPOLDA - INSURANCE TYPE (#2.312) sub-file of PATIENT (#2) IEN if  ; not adding new entry (Optional)  ; IBMVINS - Type for INSURANCE (Required)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBMVGRP - Type for GROUP (Optional)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBMVPOL - Type for POLICY (Optional)  ; 1=Merge, 2=Overwrite, 3=Replace, 4=Unsupported  ; IBNEWINS - Add new INSURANCE COMPANY flag (non-zero means add)  ; IBNEWGRP - Add new GROUP INSURANCE PLAN flag (non-zero means add)  ; IBNEWPOL - Add new patient insurance policy (non-zero means add)  ; IVMREPTR - IVM REASONS FOR NOT UPLOADING (#301.91) IEN (Optional)  ;  ;OUTPUT  ; RESULT - Returned Parameter Array with IENS of new entries and/or   ; errors/warning.  ; RESULT(0) = -1^error message  ; RESULT(0) = 0 -Move worked  ; RESULT(0) = 0 ^ warning message ^ warning message ^   ; warning message ^ warning message  ; - Move worked but there may be zero to 4 warning messages  ; RESULT(1) = "IBINSDA^" IEN of new Insurance Company (#36) File  ; RESULT(1,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 36.  ; RESULT(2) = "IBGRPDA^" IEN of new GROUP INSURANCE PLAN (#355.3)  ; File  ; RESULT(2,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 355.3.  ; RESULT(3) = "IBPOLDA^" IEN of new INSURANCE TYPE (#2.312) sub-file  ; of PATIENT (#2) IEN  ; RESULT(3,"ERR",#) - Array with any FM errors when data updated  ; from file 355.33 to 2.312.  ; RESULT(4) Contains the results of the call to UPDPOL^IBCNICB which  ; is used to update a new group into an existing patient  ; policy entry when applicable.  ; RESULT(4) =-1^error message   ; RESULT(4) =0 ^ message that process was successful or not required  ;  N IBSUPRES,IBUFSTAT,IBX  I '$D(IBELIG) S IBELIG=0  ;Set IBSUPRES to suppress screen I/O within ACCEPT  S IBSUPRES=1,IBUFSTAT=$P($G(^IBA(355.33,$G(IBBUFDA),0)),U,4)  ;  S RESULT(0)="-1^INSURANCE BUFFER (#355.33) IEN required" Q:'$G(IBBUFDA)  S RESULT(0)="-1^INSURANCE BUFFER ENTRY PREVIOUSLY PROCESSED"  Q:"~A~R~"[("~"\_IBUFSTAT\_"~")  S RESULT(0)="-1^INSURANCE BUFFER ENTRY STATUS SHOULD BE ENTERED"  Q:IBUFSTAT'="E"  S RESULT(0)="-1^PATIENT (#2) IEN required" Q:'$G(DFN)  S IBINSDA=$G(IBINSDA),IBGRPDA=$G(IBGRPDA),IBPOLDA=$G(IBPOLDA)  S IBMVINS=$G(IBMVINS,2),IBMVGRP=$G(IBMVGRP,2),IBMVPOL=$G(IBMVPOL,2)  S IBNEWINS=$G(IBNEWINS),IBNEWGRP=$G(IBNEWGRP),IBNEWPOL=$G(IBNEWPOL)  ;  S RESULT(0)="-1^Passed INSURANCE COMPANY (#36) entry doesn't exist"  I +IBINSDA,$G(^DIC(36,IBINSDA,0))="" Q  S RESULT(0)="-1^Passed GROUP INSURANCE PLAN (#355.3) entry doesn't exist"  I +IBGRPDA,$G(^IBA(355.3,IBGRPDA,0))="" Q **;2.6.6.1**  **S RESULT(0)="-1^Unable to add new INSURANCE COMPANY (#36) – See your supervisor"  I +IBNEWINS,’$D(^XUSEC(“IB INSURANCE COMPANY EDIT”,DUZ)) Q ;2.6.6.2**  **S RESULT(0)="-1^Unable to add new GROUP INSURANCE PLAN (#355.3) – See your supervisor"  I +IBNEWGRP,’$D(^XUSEC(“IB GROUP PLAN EDIT”,DUZ)) Q**  S RESULT(0)="-1^Passed Patient INSURANCE TYPE (#2.312) entry doesn't exist"  I +IBPOLDA,$G(^DPT(DFN,.312,IBPOLDA,0))="" Q  S RESULT(0)="-1^Passed GROUP INSURANCE PLAN (#355.3) entry points to different INSURANCE COMPANY (#36) entry"  I +IBGRPDA,+IBINSDA,+$G(^IBA(355.3,IBGRPDA,0))'=IBINSDA Q  S RESULT(0)="-1^Individual Policy Patient required to be Patient DFN when Group Insurance Plan is not Group Policy"  I +IBGRPDA S IBX=$G(^IBA(355.3,IBGRPDA,0)) I $P(IBX,U,2)=0,+$P(IBX,U,10),$P(IBX,U,10)'=DFN Q  ;  ;If existing GROUP INSURANCE PLAN (#355.3) entry is being changed from   ;a group plan to individual plan with other subscribers, send error   ;message and abort update  S RESULT(0)="-1^Can't change GROUP INSURANCE PLAN from Group Plan to Individual Plan when there are subscribers"  I +IBGRPDA,$P(IBX,U,2)=1,$P($G(^IBA(355.33,+$G(IBBUFDA),40)),U,1)'=1,$$SUBS^IBCNSJ(IBINSDA,IBGRPDA)>1 Q  ;  D PROCESS^IBCNBAR  Q  ;.  . |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNHHLO | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.7.2, 2.6.7.3 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBMN  IBCNHUT2  IBCNSC1  IBCNSJ3 | | | | | INIT^HLFNC2  GENERATE^HLMA  FM71^IBCNHUT2  R36^IBCNHUT2 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  SEND(INSCO) ;INSCO: IEN of Insurance Company record to send  Q:+$P($G(^IBE(350.9,1,70)),U,1)'=1 ;abort if secret HL7 flag isn't set  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  SEND(INSCO) ;INSCO: IEN of Insurance Company record to send  Q:+$P($G(^IBE(350.9,1,70)),U,1)'=1 ;abort if secret HL7 flag isn't set  **; 2.6.7.2 - Don't send HPID query if the EDI Transmit flag'="YES-LIVE"  ; 2.6.7.3 - Don't send HPID query if the Type of Coverage is: Tort/Feasor, Prescription Only, Medicaid, Medi-Cal, Disability Income Insurance, Workers’  ; Compensation, Special Class Coverage.**  .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNHUT1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.7.2, 2.6.7.3 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBMN  IBCNERPD  IBCNHUT2  IBCNSC01  IBCNSC1  IBCNSGE  IBCNSJ3  IBJTCA1  IBJTRX  IBCNPEV | | | | | FM36^IBCNHUT2 | | | |
| **Data Dictionary (DD) References** | 36,3.01 and 36,.13 | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| IBCNHUT1 ;ALB/GEF - HPID/OEID UTILITIES ;11-MAR-14  ;;2.0;INTEGRATED BILLING;\*\*519,521\*\*;21-MAR-94;Build 33  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; this routine contains various utilities for the HPID project.  Q  ; HOD(ID,INS,IBHD) ; function to determine if the data is an HPID, an OEID, or an invalid ID  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| IBCNHUT1 ;ALB/GEF - HPID/OEID UTILITIES ;11-MAR-14  ;;2.0;INTEGRATED BILLING;\*\*519,521\*\*;21-MAR-94;Build 33  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; this routine contains various utilities for the HPID project.  Q  ;  **CHKFLDS() ;2.6.7.2 Check that TRANSMIT ELECTRONICALLY field (36,#3.01)="YES"  ;2.6.7.3 Check TYPE OF COVERAGE field is not: Tort/Feasor, Prescription Only,   ; Medicaid, Medi-Cal, Disability Income Insurance,  ; Workers' Compensation, Special Class Coverage.  N EDIXMIT,TYPCOVRG  S EDIXMIT=[Code to verify EDI Transmit flag='Yes']  S TYPCOVRG=[Code to verify Type of Coverage is not above coverages]  Q EDIXMIT&TYPCOVRG**  ; HOD(ID,INS,IBHD) ; function to determine if the data is an HPID, an OEID, or an invalid ID  **.**  **.** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNUT | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.2.1 | | | | | | | | |
| **Related Options** |  | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| None | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCNUT ;ALB/FA - PATIENT POLICY COMMENT HISTORY ;15-APR-2015  ;;2.0;INTEGRATED BILLING;\*\*549\*\*;21-MAR-94;Build 120  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; General purpose eInsurance Utilities  ;  Q  ; DODUPT(DA,DOD) ;EP  ; Input Transform method to update the expiration dates of all active plans   ; for the specified patient when the patient's Date of Death is entered  ; (2,.351)  ; Input: DA - IEN of the patient  ; DOD - Date of Death input by the user  ;   ; Add code here to loop through the all of the patient's active insurance  ; plans and set the expiration date to DOD+1  Q  ;** | | | | | | | | | |

##### Protocols

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNB ENTRY SCREEN MENU | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.4.2 | | | | | | | | | | |
| **Associated Protocols** | IBCNB ENTRY EDIT INSURANCE  IBCNB ENTRY EDIT ALL  IBCNB ENTRY EDIT GROUP  IBCNB ENTRY EDIT POLICY  IBCNB FAST EXIT  **~~IBCNB ENTRY RESPONSE REPORT~~**  IBCNB EXPAND BENEFITS  IBCNB ENTRY ESCALATE | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | VALM | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| K IBFASTXT | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| I $G(IBFASTXT)=1 S VALMBCK="Q" | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

#### System Feature: Insurance Comments

RSD 2.6.9.1 Insurance Comments (History) - Within the Patient Insurance Info View/Edit option the action (AC) Add comments shall be renamed to (PT) Pt Policy Comments. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT).

RSD 2.6.9.2 Insurance Comments (History) - Within the Pt Policy Comments screen, a user shall be able to add a new comment, delete a comment (following FY14 rules and restrictions), and edit a comment (following FY14 rules and restrictions). Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT), new action to edit a comment.

RSD 2.6.9.3 Insurance Comments (History) – Within the Pt Policy Comments screen, a user shall be able to create a new comment, which includes populating the new 5 fields: 'PERSON CONTACT', 'METHOD OF CONTACT', 'CONTACT PHONE #', 'CALL REFERENCE NUMBER', 'AUTHORIZATION NUMBER'. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT), new action to add a comment.

RSD 2.6.9.4 Insurance Comments (History) - After selecting a comment to expand from within the Pt Policy Comments screen, a user shall be able to view the following data: 'PERSON CONTACT', 'METHOD OF CONTACT', 'CONTACT PHONE #', 'CALL REFERENCE NUMBER', 'AUTHORIZATION NUMBER', 'COMMENT'. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT), followed by new action to expand a comment. \*Expand command is to be consistent with RSD 2.6.9.18.

RSD 2.6.9.5 Insurance Comments (History) - Within the Pt Policy Comments screen, a user shall be able to expand a comment in order to view additional data. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT), new action to expand a comment. \*Expand command is to be consistent with RSD 2.6.9.18.

RSD 2.6.9.6 Insurance Comment (History) - The Insur. Contact Inf. (IC) action shall be dropped from the Patient Insurance Info View/Edit option. Path: Patient Insurance Info View\Edit option (PI), remove action Insur. Contact Inf. (IC).

RSD 2.6.9.7 Insurance Comment (History) - Within the Patient Insurance Info View/Edit option, under the VP action, the 'Comment -- Group Plan' section shall be moved to be below 'Plan Coverage Limitations' section and above the 'Comment -- Patient Policy' section. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP).

RSD 2.6.9.8 Insurance Comment (History) - Within the Patient Insurance Info View/Edit option, under the VP action, the most recent two patient policy comments shall be displayed. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP).

RSD 2.6.9.9 Insurance Comment (History) - Within the Patient Insurance Info View/Edit option, under the VP action, 'Insurance Contact (last)' section shall be removed. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP).

RSD 2.6.9.10 Insurance Comment (History) - Within the Patient Insurance Info View/Edit option, under the VP action, a new action Group Plan Comments (GC) shall be added which will call existing code/prompts. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), new action Group Plan Comments (GC).

RSD 2.6.9.11 Insurance Comment (History) - Within the Pt Policy Comments screen, the following elements shall be displayed in a ListMan screen: Date comment was entered (descending order, date only), user name of person who entered the comment, method of contact, person contacted, and the first several characters of the comment (approx. 70-75). Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT). \* Comment should display on its own line without wrapping.

RSD 2.6.9.12 Insurance Comment (History) - Within the Pt Policy Comments screen, the comments shall be displayed in a ListMan screen in descending order based on the Date the comment was entered. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT).

RSD 2.6.9.13 Insurance Comment (History) - Within the Pt Policy Comments screen, the user shall have the ability to do a Keyword search (contains xyz, not case sensitive) that checks the following 6 fields for that keyword; where a warning will be displayed to the user mentioning that it is an abbreviated list of comments so it may not contain all comments. Fields to be searched: 'PERSON CONTACT', 'METHOD OF CONTACT', 'CONTACT PHONE #', 'CALL REFERENCE NUMBER', 'AUTHORIZATION NUMBER', 'COMMENT'. Path: Patient Insurance Info View\Edit option (PI), followed by View Policy action (VP), followed by Pt Policy Comments (PT), followed by Search List (SL).

RSD 2.6.9.14 Insurance Comments (CT) - Within the claims tracking screens, a new action Pt Policy Comments (PT) shall be added following the defined menu path. Path: For each of the Claims Tracking Edit options (CT), followed by View Pat. Ins. (VP) action, followed by View Policy Info (VP) action, the new action (PT) shall be added here.

RSD 2.6.9.15 Insurance Comments (CT) - Within the claims tracking screens, a new action Pt Policy Comments (PT) shall provide the user with the ability to display the history of the patient's policy's comments. Path: For each of the Claims Tracking Edit options (CT), followed by View Pat. Ins. (VP) action, followed by View Policy Info (VP) action, the new action (PT) shall be added here.

RSD 2.6.9.16 Insurance Comments (CT) - Within the claims tracking screens, in the new action Pt Policy Comments (PT), the user shall have the ability to display as view only the patient's policy's comments. Path: For each of the Claims Tracking Edit options (CT), followed by View Pat. Ins. (VP) action, followed by View Policy Info (VP) action, followed by the new Pt Policy Comments action (PT).

RSD 2.6.9.17 Insurance Comments (CT) - Within the Patient Policy comment screen, the user shall have the ability to jump back to the Claims Tracking screen using normal Fileman/VistA methodology of "^" and an EXIT action.

RSD 2.6.9.18 Insurance Comments (CT) - Within the Pt Policy Comments screen, a user shall be able to expand a comment in order to view additional data. Path: For each of the Claims Tracking Edit options (CT), followed by View Pat. Ins. (VP) action, followed by View Policy Info (VP) action, followed by the new Pt Policy Comments action (PT), followed by Expand Entry action. \*Expand command is to be consistent with RSD 2.6.9.5.

RSD 2.6.9.19 Insurance Comments (TPJI) - Within the claims tracking screens, a new action Pt Policy Comments (PT) shall be added following the defined menu path. Path: Patient Insurance (PI) action, followed by Policy (VP) action, the new action shall be added here. Path: Third Party Joint Inquiry (TPJI), followed by Patient Insurance (PI) action, followed by Policy (VP) action, followed by the new Pt Policy Comments action (PT). \*Expand command is to be consistent with RSD 2.6.9.5.

RSD 2.6.9.20 Insurance Comments (TPJI) - Within the Patient Policy comment screen in TPJI, the user shall have the ability to display as view only (in detail) all patient policy comments upon individual selection and expansion. Path: Third Party Joint Inquiry (TPJI), followed by Patient Insurance (PI) action, followed by Policy (VP) action, followed by the new Pt Policy Comments action (PT), followed by Expand Entry action. \*Expand command is to be consistent with RSD 2.6.9.5.

#### Design Element Tables (Insurance Comments)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNCH | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.9.2, 2.6.9.3, 2.6.9.4, 2.6.9.5, 2.6.9.8, 2.6.9.9, 2.6.9.11, 2.6.9.12, 2.6.9.13, 2.6.9.16, 2.6.9.18, 2.6.9.20 | | | | | | | | |
| **Related Options** | Patient Insurance Info View/Edit [IBCN PATIENT INSURANCE] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | | |
|  | None | | | | | None | | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | | |
| **Related Protocols** | IBCNCH COMMENT HISTORY MENU, IBCNCH ADD COMMENT, IBCNCH EXPAND COMMENT, IBCNCH EDIT COMMENT, IBCNCH SEARCH LIST, IBCNCH DELETE COMMENT | | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local | |
| **Input Attribute Name and Definition** | None | | | | | | | | | |
| **Output Attribute Name and Definition** | None | | | | | | | | | |
| **Current Logic** | | | | | | | | | | |
|  | | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | | |
| IBCNCH ;ALB/FA - PATIENT POLICY COMMENT HISTORY ;05-MAR-2015  ;;2.0;INTEGRATED BILLING;\*\*6,28,43,52,85,251,363,371,416,497\*\*;21-MAR-94;Build 120  ;;Per VA Directive 6402, this routine should not be modified. EN(DFN,IBBPOL) ;EP   ; Main entry point  ; Input: DFN - IEN of the patient  ; IBPPOL - ^DPT(DFN,.312,PIEN,0) Where PIEN is the IEN of the  ; selected patient policy  K VALMQUIT  ;S IBTOP="IBCNCH"  S DFN=7168861  S IBPPOL="1^2^7168861^1^8^DSLJFLKSD^00123^2960312^^v^^2900922^^^^^^^KDKDK^01^CROWE,JANET^324^^"  D EN^VALM("IBCNCH POLICY COMMENT HISTORY")  Q  ; HDR ;EP  ; Build the listman template header information  ; Input: DFN - IEN of the patient  ; IBPPOL - ^DPT(DFN,.312,PIEN,0) Where PIEN is the IEN of the  ; selected patient policy  N WW,XX,YY,ZZ  S XX=$E($P(^DPT(DFN,0),"^",1),1,20)\_" "\_$P($$PT^IBEFUNC(DFN),"^",2)  S ZZ=$$GET1^DIQ(2,DFN\_",",.03),XX=XX\_" "\_ZZ  S VALMHDR(1)="Policy Comment History for: "\_XX  S ZZ=$G(^DPT(DFN,.312,+$P(IBPPOL,"^",4),0))  S WW=$P($G(^IBA(355.3,+$P(ZZ,"^",18),0)),"^",11)  S YY=$E($P($G(^DIC(36,+ZZ,0)),"^",1),1,20)\_" Insurance Company"  S XX="\*\* Plan Currently "\_$S(WW:"Ina",1:"A")\_"ctive \*\*"  S VALMHDR(2)=$$SETSTR^VALM1(XX,YY,48,29)  Q  ; INIT ;EP  ; Initialize the listman template  ; Input: DFN - IEN of the patient  ; IBPPOL - ^DPT(DFN,.312,PIEN,0) Where PIEN is the IEN  ; of the selected patient policy  ; Output: ^TMP("IBCNCH",$J) - Body lines to display for specified template  ; ^TMP($J,"IBCNCHIX") - Index of displayed comments (see GETCOMS)  K ^TMP("IBCNCH",$J),^TMP($J,"IBCNCHIX")  D BLD(DFN,IBPPOL)  Q  ; BLD(DFN,IBPPOL) ; Build the listman template body  ; Input: DFN - IEN of the patient  ; IBPPOL - ^DPT(DFN,.312,PIEN,0) Where PIEN is the  ; IEN of the selected patient policy  ; ^TMP($J,"IBCNCHIX",CNT) - See GetCOMS  N CNT,LINE,LN,XX  D GETCOMS(DFN,IBPPOL)  S VALMCNT=0,LINE="",CNT=""  F  D  Q:CNT=""  . S CNT=$O(^TMP($J,"IBCNCHIX",CNT))  . Q:CNT=""  . I CNT'=1 D  . . S VALMCNT=VALMCNT+1  . . D SET^VALM10(VALMCNT,"",VALMCNT)  . S VALMCNT=VALMCNT+1,LINE=$$SETL("",CNT,"",1,4)  . S XX="User: "\_$P(^TMP($J,"IBCNCHIX",CNT),"^",2)  . S LINE=$$SETL(LINE,XX,"",6,37)  . D SET^VALM10(VALMCNT,LINE,VALMCNT)  . S XX="Dt Ent: "\_$P(^TMP($J,"IBCNCHIX",CNT),"^",1)  . S LINE=$$SETL(LINE,XX,"",39,62)  . D SET^VALM10(VALMCNT,LINE,VALMCNT)  . S XX="Method: "\_$P(^TMP($J,"IBCNCHIX",CNT),"^",4)  . S LINE=$$SETL(LINE,XX,"",66,80)  . D SET^VALM10(VALMCNT,LINE,VALMCNT)  . S VALMCNT=VALMCNT+1  . S XX=$E(^TMP($J,"IBCNCHIX",CNT,1),1,132)  . S LINE=$$SETL("",XX,"",6,80)  . D SET^VALM10(VALMCNT,LINE,VALMCNT)  Q  ; GETCOMS(DFN,IBPPOL) ; Retieves the policy comments for the selected  ; patient and policy in most recent date order  ; Input: DFN - IEN of the patient  ; IBPPOL - ^DPT(DFN,.312,PIEN,0) Where PIEN is the  ; IEN of the selected patient policy  ; Output: ^TMP($J,"IBCNCHIX",CNT) - A1^A2^A3^A4^A5^A6^A7  ; ^TMP($J,"IBCNCHIX",CNT,1) - B1  ; .  ; .  ; ^TMP($J,"IBCNCHIX",CNT,n) - Bn  ; Where:  ; CNT - Comment selection # (comments ordered by  ; (date entered)  ; A1 - External date/time the comment was entered  ; A2 - External User Name of the user who entered  ; the comment   ; A3 - Person Contacted  ; A4 - Method of Contact  ; A5 - Contact Phone #  ; A6 - Call Reference #  ; A7 - Authorization #  ; B1 - First line of comment  ; Bn - Last line of comment  S ^TMP($J,"IBCNCHIX",1)="03/01/15@09:27am^ALTMAN,FRED^ALTMAN,FRED^PHONE^207-744-9048^1234^5431"  S ^TMP($J,"IBCNCHIX",1,1)="THIS IS THE FIRST LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",1,2)="THIS IS THE SECOND LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",1,3)="THIS IS THE THIRD LINE OF THE COMMENT"  ;  S ^TMP($J,"IBCNCHIX",2)="02/01/15@10:20am^GRAHAM,STEVEN^GRAHAM,STEVEN^PHONE^555-555-2121^1234^5431"  S ^TMP($J,"IBCNCHIX",2,1)="THIS IS THE FIRST LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",2,2)="THIS IS THE SECOND LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",2,3)="THIS IS THE THIRD LINE OF THE COMMENT"  ;  S ^TMP($J,"IBCNCHIX",3)="01/22/15@01:32pm^BARRETT,ORLANDO^BARRETT,ORLANDO^PHONE^555-555-21212^1234^5431"  S ^TMP($J,"IBCNCHIX",3,1)="THIS IS THE FIRST LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",3,2)="THIS IS THE SECOND LINE OF THE COMMENT"  S ^TMP($J,"IBCNCHIX",3,3)="THIS IS THE THIRD LINE OF THE COMMENT"  Q  ; SETL(LINE,DATA,LABEL,COL,LNG) ; Creates a line of data to be set into the body  ; of the worklist  ; Input: LINE - Current line being created  ; DATA - Information to be added to the end of the current line  ; LABEL - Label to describe the information being added  ; COL - Column position in line to add information add  ; LNG - Maximum length of data information to include on the line  ; Returns: Line updated with added information  S LINE=LINE\_$J("",(COL-$L(LABEL)-$L(LINE)))\_LABEL\_$E(DATA,1,LNG)  Q LINE  ; ADDCOM  ;EP  ; Protocol action to Add a new Patient Policy Comment **;2.6.9.16, 2.6.9.20 - Add code to disable if coming from CT or TJPI**   N COMNUM  S VALMBCK="R"  S COMNUM=$$SELCOM(1,"Select Comment to expand","","IBCNCHIX")  Q:COMNUM="" **; Add call ^DIE with new fields here**  Q  ; DELETE  ;EP  ; Protocol action to Delete a Patient Policy Comment  **;2.6.9.16, 2.6.9.20 - Add code to disable if coming from CT or TJPI**  **; Add FY14 code to check if it’s ok to delete here**   N COMNUM,DA,DIK  S VALMBCK="R"  S COMNUM=$$SELCOM(1,"Select Comment to expand","","IBCNCHIX")  Q:COMNUM=""  **; Set DIK to the correct multiple here**   S DA=VAIEN,DIK=""   D ^DIK ; Delete the Vendor Agreement  Q  ; EDIT  ;EP  ; Protocol action to Edit a Patient Policy Comment Fields  **;2.6.9.16, 2.6.9.20 - Add code to disable if coming from CT or TJPI**  N COMNUM,DA,DIC,DIE,DO,DR,DTOUT,X,Y  S VALMBCK="R" **; Add FY14 code to check if it’s ok to edit here**   S COMNUM=$$SELCOM(1,"Select Comment to expand","","IBCNCHIX")  Q:COMNUM=""  S DIE=2.312,DA=XXX **; Set DR to the correct fields here**   S DR="4:16"  D ^DIE  Q  ; EXPCOM ;EP **; 2.6.9.18**  ;Protocol action to expand a selected Patient Policy Comment  ; Input: ^TMP($J,"IBCNCHIX") - Index of displayed lines of the Comment   ; History Worklist  ; Output: All Policy Comment History fields displayed  N COMNUM,IX,XX  S VALMBCK="R"  S COMNUM=$$SELCOM(1,"Select Comment to expand","","IBCNCHIX")  Q:COMNUM=""  S XX=^TMP($J,"IBCNCHIX",COMNUM)  W @IOF,?25,"Patient Policy Comment #",COMNUM  W !,"Entered By: ",$P(XX,"^",2),?45,"Date/Time Entered: ",$P(XX,"^",1)  W !,"Person Contacted: ",$P(XX,"^",3),?45,"Contact Method: ",$P(XX,"^",4)  W !,"Contact Phone #: ",$P(XX,"^",5)  W !,"Call Reference #: ",$P(XX,"^",6),?45,"Authorization #: ",$P(XX,"^",7)  W !!,"Comment:"  S IX=""  F  D  Q:IX=""  . S IX=$O(^TMP($J,"IBCNCHIX",COMNUM,IX))  . Q:IX=""  . W !,^TMP($J,"IBCNCHIX",COMNUM,IX)  D PAUSE^VALM1  Q  ; SEARCH  ;EP  ; Protocol action to Search Patient Policy Comments for selected text **; Add code to let the user enter the search text and search for it here**   Q  ; HELP ;EP  ; Display the listman template help  S X="?" D DISP^XQORM1 W !!  Q  ; SELCOM(FULL,PROMPT,DLINE,WLIST) ;EP  ; Select Entry(s) to perform an action upon  ; Input: FULL - 1 - full screen mode, 0 otherwise  ; PROMPT - Prompt to be displayed to the user  ; WLIST - Worklist, the user is selecting from  ; ^TMP($J,"IBCNCHIX") - Index of displayed lines of the Comment   ; History Worklist  ; Output: DLINE - Comma delimitted list of Line #(s) of the   ; selected entries  ; Returns: Select Comment #  ; Error message if invalid selection  N DIROUT,DIRUT,DTOUT,DUOUT,END,START,X,Y  S:'$D(WLIST) WLIST="IBCNCHIX"  S START=1,END=$O(^TMP($J,WLIST,""),-1)  D:FULL FULL^VALM1  S DLINE=$P($P($G(XQORNOD(0)),"^",4),"=",2) ; User selection with action  S DLINE=$TR(DLINE,"/\; .",",,,,,") ; Check for multi-selection  ;  I DLINE["," D  Q ""                            ; Invalid multi-selection  . W !,\*7,">>>> Only single entry selection is allowed"  . S DLINE=""  . K DIR  . D PAUSE^VALM1  S:DLINE="" DLINE=$$SELENTRY(PROMPT,START,END)  Q:DLINE="" ""  Q DLINE  ; SELENTRY(PROMPT,START,END) ; select a comment  ; Input: PROMPT - Prompt to be displayed to the user  ; START - Start comment # that can be selected  ; END - Ending comment # that can be selected  ; Returns: Selected Comment # or "" if not selected  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DIR(0)="NO^"\_START\_":"\_END\_":0"  S DIR("A")=PROMPT  D ^DIR K DIR  Q X  ; EXIT ;EP  ; Exit the listman template  K ^TMP("IBCNCH",$J),^TMP($J,"IBCNCHIX")  D CLEAR^VALM1  Q | | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNSP3 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.9.1, 2.6.9.10 | | | | | | | | |
| **Related Options** | Patient Insurance Info View/Edit [IBCN PATIENT INSURANCE] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCEOB01  IBCNSJ  IBCNSM3  IBCNSM32  IBCNSP1  IBCN1 | | | | | IBCNSM4  COVERED^IBCNSM31  LOCKED^IBTRCD1  BLD^IBCNSP  BLANK^IBCNSP  BLD^IBCNSM  EXPAND^IBTRE | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBCNSP POLICY MENU, IBCNSP ADD COMMENT | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .   ; AC ; -- Add Comment  D FULL^VALM1 W !!  N IBDIF,DA,DR,DIE,DIC,X,Y  D SAVEPT(DFN,IBCDFN)  W !!,"You may now enter a brief comment about this patient's policy"  D VARS  L +^DPT(DFN,.312,+$P($G(IBPPOL),"^",4)):5 I '$T D LOCKED^IBTRCD1 G ACQ  S DR="1.08" D ^DIE  D COMPPT(DFN,IBCDFN) I IBDIF D UPDATPT(DFN,IBCDFN)  L -^DPT(DFN,.312,+$P($G(IBPPOL),"^",4))  W !!,"You may now enter comments about this Group Plan that pertains to all Patients"  L +^IBA(355.3,+IBCPOL):5 I '$T D LOCKED^IBTRCD1 G ACQ  S DIE="^IBA(355.3,",DA=IBCPOL,DR="11" D ^DIE  D BLD^IBCNSP  L -^IBA(355.3,+IBCPOL) ACQ S VALMBCK="R" Q  ;  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  .   ;  **AC ; -- Add Comment  ; 2.6.9.1 Remove Method**  **D FULL^VALM1 W !!  N IBDIF,DA,DR,DIE,DIC,X,Y  D SAVEPT(DFN,IBCDFN)  W !!,"You may now enter a brief comment about this patient's policy"  D VARS  L +^DPT(DFN,.312,+$P($G(IBPPOL),"^",4)):5 I '$T D LOCKED^IBTRCD1 G ACQ  S DR="1.08" D ^DIE  D COMPPT(DFN,IBCDFN) I IBDIF D UPDATPT(DFN,IBCDFN)  L -^DPT(DFN,.312,+$P($G(IBPPOL),"^",4))  W !!,"You may now enter comments about this Group Plan that pertain to all Patients"  L +^IBA(355.3,+IBCPOL):5 I '$T D LOCKED^IBTRCD1 G ACQ  S DIE="^IBA(355.3,",DA=IBCPOL,DR="11" D ^DIE  D BLD^IBCNSP  L -^IBA(355.3,+IBCPOL) ACQ S VALMBCK="R" Q  ;**  **GC ;2.6.9.10 -- Group Comment Add Method**  **D FULL^VALM1**  **W !!  N IBDIF,DA,DR,DIE,DIC,X,Y  D SAVEPT(DFN,IBCDFN) D COMPPT(DFN,IBCDFN) I IBDIF D UPDATPT(DFN,IBCDFN)  L +$P($G(IBPPOL),"^",4))  W !!,"You may now enter comments about this Group Plan that pertain to all Patients"  L +^IBA(355.3,+IBCPOL):5 I '$T D LOCKED^IBTRCD1 G ACQ  S DIE="^IBA(355.3,",DA=IBCPOL,DR="11" D ^DIE  D BLD^IBCNSP  L -^IBA(355.3,+IBCPOL) ACQ S VALMBCK="R" Q**;  .  . | | | | | | | | | |

##### Templates

| **Template Name** | | IBCNCH POLICY COMMENT | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Enhancement Category** | | New | Modify | Delete | | No Change |
| **RSD** | | RSD 2.6.9.11, 2.6.9.12 | | | | |
| **Template Type** | | Sort | Input | Print | | Other |
| **Related Options** | | VALM HIDDEN ACTIONS | | | | |
| **Related Routines** | | **Routines “Called By”** | | | **Routines “Called”** | |
|  | | IBCNCH | | | EXIT^IBCNCH  HDR^IBCNCH  HELP^IBCNCH  INIT^IBCNCH | |
| Routines | Description | | | | |
| **Data Dictionary (DD) References** | None | | | | |
| **Global References** | ^TMP("IBCNCH",$J) | | | | |

##### Protocols

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNSP POLICY MENU | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.1, 2.6.9.6 | | | | | | | | | | |
| **Associated Protocols** | IBCNSP EDIT POLICY INFO  IBCNSP EDIT EFFECTIVE DATES  IBCNSP VERIFY COVERAGE  IBCNSP SUBSCRIBER UPDATE  IBCNSP EMPLOYER INFO FOR CLAIMS  IBCNSP UR INFO  IBCNSP ANNUAL BENEFITS  IBCNSP BENEFITS USED  **~~IBCNSP INSURANCE CONTACT INF~~**  **IBCNSP ADD GROUP COMMENT**  IBCNSJ INACTIVATE PLAN  IBCNSJ CHANGE PLAN  IBCNSJ EDIT COVERAGE LIMITS  **~~IBCNSP ADD COMMENT~~**  **IBCNSP PATIENT POLICY COMMENTS**  IBCNSP EDIT ALL  IBCNE VP VIEW EXP ELIG BEN SCREEN  IBCNS EXIT  VALM BLANK 1 | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | VALM | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| I $G(IBFASTXT)=1 S VALMBCK="Q" | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNSV POLICY MENU | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.14, 2.6.9.15 | | | | | | | | | | |
| **Associated Protocols** | IBCNE VP VIEW EXP ELIG BEN SCREEN  **IBCNCH PATIENT POLICY COMMENTS**  IBCNS EXIT | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | VALM | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| I $G(IBFASTXT)=1 S VALMBCK="Q" | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBJT NS VIEW EXP POL MENU | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.19 | | | | | | | | | | |
| **Associated Protocols** | IBJT ACTIVE LIST SCREEN SKIP  **IBCNCH PATIENT POLICY COMMENTS** | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | VALM | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| I $G(IBFASTXT)=1 S VALMBCK="Q" | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH COMMENT HISTORY MENU | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.2, 2.6.9.3, 2.6.9.4, 2.6.9.5, 2.6.9.13, 2.6.9.17 | | | | | | | | | | |
| **Associated Protocols** | IBCNCH ADD COMMENT  IBCNCH EDIT COMMENT  IBCNCH DELETE COMMENT  IBCNCH EXPAND COMMENT  IBCNCH SEARCH LIST  IBCNS EXIT | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | VALM | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| I $G(IBFASTXT)=1 S VALMBCK="Q" | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH ADD COMMENT | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.3 | | | | | | | | | | |
| **Associated Protocols** | None | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | IBCNCH | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| D ADDCOM^IBCNCH | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH EDIT COMMENT | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.3 | | | | | | | | | | |
| **Associated Protocols** | None | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | IBCNH | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| D EDIT^IBCNCH | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH EXPAND COMMENT | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.3 | | | | | | | | | | |
| **Associated Protocols** | None | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | IBCNCH | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| D EXPCOM^IBCNCH | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH DELETE COMMENT | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.3 | | | | | | | | | | |
| **Associated Protocols** | None | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | IBCNCH | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| D DELETE^IBCNCH | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** | IBCNCH SEARCH LIST | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **RTM** | 2.6.9.3 | | | | | | | | | | |
| **Associated Protocols** |  | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | Process  Menu | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** | IBCNCH | | | | | | | | | | |
| Current Entry Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Entry Action Logic (Changes are in bold) | | | | | | | | | | | |
| D SEARCH^IBCNCH | | | | | | | | | | | |
| Current Exit Action Logic | | | | | | | | | | | |
| None | | | | | | | | | | | |
| Modified Exit Action Logic (Changes are in bold) | | | | | | | | | | | |
| None | | | | | | | | | | | |

#### System Feature: IV Site Parameters

RSD 2.6.10.1 IV Site Parameters - The eIV Site Parameter shall display the MEDICARE PAYER (#350.9, 51.25) field.

RSD 2.6.10.2 IV Site Parameters - The eIV Site Parameter shall allow the user to edit the MEDICARE PAYER (#350.9, 51.25) field.

RSD 2.6.10.3 IV Site Parameters - The eIV Site Parameter shall display as view only the RETRY FLAG (#350.9, 51.26) field.

RSD 2.6.10.4 IV Site Parameters - The eIV Site Parameter shall display as view only the TIMEOUT DAYS (#350.9, 51.05) field.

RSD 2.6.10.5 IV Site Parameters - The eIV Site Parameter shall display as view only the TIMEOUT MAILMAN MSG (#350.9, 51.07) field.

RSD 2.6.10.6 IV Site Parameters - The eIV Site Parameter shall display as view only the NUMBER RETRIES (#350.9, 51.06) field.

RSD 2.6.10.7 IV Site Parameters - The eIV Site Parameter shall display as view only the DEFAULT SERVICE TYPE CODE 1 (#350.9, 60.01) field.

RSD 2.6.10.8 IV Site Parameters - The eIV Site Parameter shall display the HMS DIRECTORY (#350.9, 13.01) field.

RSD 2.6.10.9 IV Site Parameters - The eIV Site Parameter shall allow the user to edit the HMS DIRECTORY (#350.9, 13.01) field.

RSD 2.6.10.10 IV Site Parameters - The eIV Site Parameter shall display the EII ACTIVE (#350.9, 13.02) field.

RSD 2.6.10.11 IV Site Parameters - The eIV Site Parameter shall allow the user to edit the EII ACTIVE (#350.9, 13.02) field.

RSD 2.6.10.12 IV Site Parameters - The eIV Site Parameter shall display as view only the HL7 MAXIMUM NUMBER (#350.9, 51.15) field.

RSD 2.6.10.13 IV Site Parameters - The eIV Site Parameters shall no longer reference (no display or edit) the CONTACT PERSON (#350.9, 51.16) field.

RSD 2.6.10.14 IV Site Parameters - The eIV Site Parameters shall no longer reference (no display or edit) the phone number associated with the CONTACT PERSON (#350.9, 51.16) field.

RSD 2.6.10.15 IV Site Parameters - The eIV Site Parameters shall no longer reference (no display or edit) the email address associated with the CONTACT PERSON (#350.9, 51.16) field.

RSD 2.6.10.16 IV Site Parameters - The CONTACT PERSON (#350.9, 51.16) field shall be dropped from the IB Site Parameters file (#350.9).

RSD 2.6.10.17 IV Site Parameters - eIV Table updates shall be modified so that FSC may use a non-payer table update to control the following VistA fields:

HL7 MAXIMUM NUMBER (#350.9, 51.15)

MAXIMUM EXTRACT NUMBER (#350.9002, .05) for the buffer extract

MAXIMUM EXTRACT NUMBER (#350.9002, .05) for the appt extract

eIV Master Switches (two new fields to be added to file #350.9. Field #s TBD during design. Field names subject to change during design.)

RSD 2.6.10.18 IV Site Parameters – A new field shall be added to the IV Site Parameters (eIV Master Switch (real time)), as view only, that will be used to control whether any eIV real time 270 transactions can be created and transmitted to the Eligibility Communicator (EC).

RSD 2.6.10.19 IV Site Parameters – Within the MCCR Site Parameters option, the insurance verification action resulting display page shall be renamed IV Site Parameters.

RSD 2.6.10.20 IV Site Parameters – The IV Site Parameter shall display as view only the Failure Mailman Msg (#350.9, 51.2) field.

RSD 2.6.10.21 IV Site Parameters – The IV Site Parameter shall display as view only the Messages Mailgroup (#350.9, 51.04) field.

RSD 2.6.10.22 IV Site Parameters – A new field shall be added to the IV Site Parameters (eIV Master Switch (extracts)), as view only, that will be used to control whether any eIV 270 transactions from the extracts can be created and transmitted to the Eligibility Communicator (EC).

#### Design Element Tables (IV Site Parameters)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBJPI | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.10.1, 2.6.10.3, 2.6.10.4, 2.6.10.5, 2.6.10.6, 2.6.10.7, 2.6.10.8, 2.6.10.9, 2.6.10.10, 2.6.10.12, 2.6.10.13, 2.6.10.14, 2.6.10.15, 2.6.10.20, 2.6.10.21, 2.6.10.22 | | | | | | | | |
| **Related Options** | MCCR Site Parameter Display/Edit [IBJ MCCR SITE PARAMETERS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | | |
|  | None | | | | | FO^IBCNEUT1  MGRP^IBCNEUT5 | | | | |
| **Routines** | **Activities** | | | | | | | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | | |
| **Related Protocols** | IBJP INS VER SCREEN | | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | NONE | | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local | |
| **Input Attribute Name and Definition** | NONE | | | | | | | | | |
| **Output Attribute Name and Definition** | NONE | | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .  BLD ; build screen array  N IBLN,IBCOL,IBWID,IBIIV,IBIIVB,IBIEN,CT,IBEX1,IBEX2,IBEX,IEN  N IBST,IBDATA,DISYS,X,STATUS,AIEN,ADATA  ;  S (IBLN,VALMCNT)=0,IBCOL=3,IBIIV=$G(^IBE(350.9,1,51))  ; -- Gen Params  S IBWID=49  S IBLN=$$SETN("General Parameters",IBLN,IBCOL,1,)  S IBLN=$$SET("Days between electronic re-verification checks: ",$P(IBIIV,U,1),IBLN,IBWID)  S IBLN=$$SET("Send daily statistical report via MailMan: ",$S($P(IBIIV,U,2):"YES",$P(IBIIV,U,2)=0:"NO",1:""),IBLN,IBWID)  I $P(IBIIV,U,2) S IBLN=$$SET("Time of day for daily statistical report: ",$P(IBIIV,U,3),IBLN,IBWID)  S IBLN=$$SET("Mail Group for eIV messages: ",$$MGRP^IBCNEUT5,IBLN,IBWID)  ;  S IBLN=$$SET("Contact Person: ",$S($P(IBIIV,U,16)'="":$$GET1^DIQ(200,$P(IBIIV,U,16)\_",",.01,"E"),1:""),IBLN,IBWID)  S IBLN=$$SET("Send MailMan message if communication problem: ",$S($P(IBIIV,U,20):"YES",$P(IBIIV,U,20)=0:"NO",1:""),IBLN,IBWID)  ;  ; Skip lines in between sections  S IBLN=$$SET("","",IBLN,0)  ;  ; -- Batch Extracts  S IBWID=43  S IBLN=$$SETN("Batch Extracts",IBLN,IBCOL,1,)  S IBLN=$$SET("Extract Selection Maximum # to","",IBLN,IBWID)  S IBLN=$$SETN(" Name On/Off Criteria Extract/Day",IBLN,IBCOL+1,,1)  ;S IBLN=$$SETN(" Extract Name On/Off Selection Criteria",IBLN,IBCOL+1,,1)  ; Loop thru extracts  S IEN=0 F  S IEN=$O(^IBE(350.9,1,51.17,IEN)) Q:'IEN  D  . S IBIIVB=$G(^IBE(350.9,1,51.17,IEN,0))  . S IBEX=+$P(IBIIVB,U,1) ; Type  . I '$F(".1.2.","."\_IBEX\_".") Q  . S IBST=$$FO^IBCNEUT1($S($P(IBIIVB,U,1)'="":$$GET1^DIQ(350.9002,IEN\_",1,",.01,"E"),1:""),14)  . S IBST=IBST\_$$FO^IBCNEUT1($S(+$P(IBIIVB,U,2):"ON",1:"OFF"),8)  . S IBEX1=$S(+$P(IBIIVB,U,3)'=0:+$P(IBIIVB,U,3),1:$P(IBIIVB,U,3))  . S IBEX2=$S(+$P(IBIIVB,U,4)'=0:+$P(IBIIVB,U,4),1:$P(IBIIVB,U,4))  . S IBST=IBST\_$$FO^IBCNEUT1($S(IBEX=1:"n/a",IBEX=2:IBEX1,IBEX=3:IBEX1\_"/"\_IBEX2,1:"ERROR"),11)  . S IBST=IBST\_$$FO^IBCNEUT1($S(+$P(IBIIVB,U,5):+$P(IBIIVB,U,5),1:$P(IBIIVB,U,5)),14)  . S IBLN=$$SET(IBST,"",IBLN,IBWID)  . Q  ;S IBLN=$$SET("","",IBLN,0)  S VALMCNT=IBLN  Q  ;  .  . |

| **Modified Logic (Changes are in bold)** |
| --- |
| ; BLD ; Creates the body of the worklist  ;  **N ELINEL,ELINER,SLINE,STARTR  S VALMCNT=0,SLINE=1  D BLDGENE(SLINE,.ELINEL) ; Build Editable General Parameters  D BLDGENNL(ELINEL,.STARTR,.ELINEL) ; Build Non-Editable Gen Param left  D BLDGENNR(STARTR,.ELINER) ; Build Non-Editable Gen Param Right  S SLINE=$S(ELINEL>ELINER:ELINEL,1:ELINER)  D BLDGENNB(SLINE,.ELINEL) ; Build Non-Editable Bottom Params  D BLDBE(ELINEL,.ELINEL) ; Build Batch Extract Gen Parameters  S VALMCNT=ELINEL-1  Q  ; BLDGENE(SLINE,ELINE) ; Build the General Editable Parameters Section  ; Input: SLINE - Starting Section Line Number  ; ELINE - Current Ending Section Line Number  ; Output: ELINE - Updated Ending Section Line Number  ;2.6.10.13 Contact Person removed  ;2.6.10.14 Contact Person Phone Number removed  ;2.6.10.15 Contact Person Email Address removed  ;  ;S ELINE=$$SET("",$J("",40),SLINE,1) ; Spacing Blank Line  S ELINE=$$SETN("General Parameters (editable)",SLINE,1,1)  ; 2.6.10.1**  **S ELINE=$$SET("Medicare Payer: ",$$GET1^DIQ(350.9,"1,",51.25),ELINE,1)  ; 2.6.10.8**  **S ELINE=$$SET(" HMS Directory: ",$$GET1^DIQ(350.9,"1,",13.01),ELINE,1)  ; 2.6.10.10**  **S ELINE=$$SET(" EII Active: ",$$GET1^DIQ(350.9,"1,",13.02),ELINE,1)  Q  ; BLDGENNL(SLINE,STARTR,ELINE) ; Build the Left portion of the General  ; Non-Editable Parameters Section  ; Input:SLINE - Starting Section Line Number  ; ELINE - Current Ending Section Line Number  ; Output: STARTR - Line to start displaying General Non-Editable Right  ; Section  ; ELINE - Updated Ending Section Line Number  ;  S ELINE=$$SET("",$J("",40),SLINE,1) ; Spacing Blank Line  S ELINE=$$SETN("General Parameters (non-editable)",ELINE,1,1)  S STARTR=ELINE                                 ; Start of Right Section  S ELINE=$$SET(" Freshness Days: ",$$GET1^DIQ(350.9,"1,",51.01),ELINE,1)  ; 2.6.10.4**  **S ELINE=$$SET(" Timeout Days: ",$$GET1^DIQ(350.9,"1,",51.05),ELINE,1)  ; 2.6.10.5**  **S ELINE=$$SET("Timeout Mailman Msg: ",$$GET1^DIQ(350.9,"1,",51.07),ELINE,1)  ; 2.6.10.7**  **S ELINE=$$SET(" Default STC: ",$$GET1^DIQ(350.9,"1,",60.01),ELINE,1)  Q  ; BLDGENNR(SLINE,ELINE) ; Build the Right portion of the General  ; Non-Editable Parameters Section  ; Input: SLINE - Starting Section Line Number  ; ELINE - Current Ending Section Line Number  ; Output: ELINE - Updated Ending Section Line Number  ;  S ELINE=SLINE  ; 2.6.10.12**  **S ELINE=$$SET("HL7 Maximum Number: ",$$GET1^DIQ(350.9,"1,",51.15),ELINE,41)  ; 2.6.10.3**  **S ELINE=$$SET(" Retry Flag: ",$$GET1^DIQ(350.9,"1,",51.26),ELINE,41)  ; 2.6.10.6**  **S ELINE=$$SET(" Number of Retries: ",$$GET1^DIQ(350.9,"1,",51.26),ELINE,41)  S ELINE=$$SET(" Mail Group: ",$$MGRP^IBCNEUT5,ELINE,41)  Q  ; BLDGENNB(SLINE,ELINE) ; Build the General Non-Editable Bottom Parameters Section  ; Input: SLINE - Starting Section Line Number  ; ELINE - Current Ending Section Line Number  ; Output: ELINE - Updated Ending Section Line Number  ;  N XX  S ELINE=$$SET("",$J("",40),SLINE,1) ; Spacing Blank Line  ; 2.6.10.20 – next 3 lines**  **S XX=$$GET1^DIQ(350.9,"1,",51.2)  S:XX="" XX="NO"  S ELINE=$$SET("Send MailMan Message if Communication Problem: ",XX,ELINE,1)  ; 2.6.10.21 – next 4 lines**  **S XX=$$GET1^DIQ(350.9,"1,",51.02)  S:XX="" XX="NO"  S XX=$$GET1^DIQ(350.9,"1,",51.02)\_" at "\_$$GET1^DIQ(350.9,"1,",51.03)  S ELINE=$$SET(" Receive MailMan Message, Daily Statistical: ",XX,ELINE,1)  ; 2.6.10.22 – next 3 lines**  **S XX=$$GET1^DIQ(350.9,"1,",51.27)  S:XX="" XX="NO"  S ELINE=$$SET("270 Master Switch Realtime: ",XX,ELINE,1)   S XX=$$GET1^DIQ(350.9,"1,",51.28)  S:XX="" XX="NO"  S ELINE=$$SET("270 Master Switch Nightly: ",XX,ELINE,1)**  **Q  ; BLDBE(SLINE,ELINE) ; Build the Batch Extract Parameters Section  ; Input: SLINE - Starting Section Line Number  ; ELINE - Current Ending Section Line Number  ; Output: ELINE - Updated Ending Section Line Number  ;  N IBEX,IBEX1,IBEX2,IBIIVB,IBST,IEN  S ELINE=$$SET("",$J("",40),SLINE,1) ; Spacing Blank Line  S ELINE=$$SET("",$J("",40),ELINE,1) ; Spacing Blank Line  S ELINE=$$SET("",$J("",40),ELINE,1) ; Spacing Blank Line  S ELINE=$$SET("",$J("",40),ELINE,1) ; Spacing Blank Line  S ELINE=$$SETN("Batch Extracts",ELINE,1,1)  S ELINE=$$SET(" Extract Selection Maximum # to","",ELINE,1)  S ELINE=$$SETN("Name On/Off Criteria Extract/Day",ELINE,1,"",1)  ;  ; Loop thru extracts  S IEN=0  F  D  Q:'IEN  . S IEN=$O(^IBE(350.9,1,51.17,IEN))  . Q:'IEN  . S IBIIVB=$G(^IBE(350.9,1,51.17,IEN,0)) ; Batch Extract multiple line  . S IBEX=+$P(IBIIVB,"^",1) ; Type  . Q:'$F(".1.2.","."\_IBEX\_".")  . S IBST=$$FO^IBCNEUT1($S($P(IBIIVB,"^",1)'="":$$GET1^DIQ(350.9002,IEN\_",1,",.01,"E"),1:""),14)  . S IBST=IBST\_$$FO^IBCNEUT1($S(+$P(IBIIVB,"^",2):"ON",1:"OFF"),8)  . S IBEX1=$S(+$P(IBIIVB,U,3)'=0:+$P(IBIIVB,"^",3),1:$P(IBIIVB,"^",3))  . S IBEX2=$S(+$P(IBIIVB,U,4)'=0:+$P(IBIIVB,"^",4),1:$P(IBIIVB,"^",4))  . S IBST=IBST\_$$FO^IBCNEUT1($S(IBEX=1:"n/a",IBEX=2:IBEX1,IBEX=3:IBEX1\_"/"\_IBEX2,1:"ERROR"),11)  . S IBST=IBST\_$$FO^IBCNEUT1($S(+$P(IBIIVB,"^",5):+$P(IBIIVB,"^",5),1:$P(IBIIVB,"^",5)),14)  . S ELINE=$$SET(IBST,"",ELINE,1)  Q  ;** |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNEHLT | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.10.17 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNEHLI | | | | | DECHL7^IBCNEHL2 SPAR^IBCNEHLU BUFF^IBCNEUT2  RSP^IBCNEUT2  SST^IBCNEUT2  MGRP^IBCNEUT5 MSG^IBCNEUT5  IBCNRHLT | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNEHLT ;DAOU/ALA - HL7 Process Incoming MFN Messages ; 09 Dec 2005 3:30 PM  ;;2.0;INTEGRATED BILLING;\*\*184,251,271,300,416,438,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process incoming MFN messages and  ; update the appropriate tables  ; EN ; Entry Point  NEW AIEN,APIEN,APP,D0,D,DESC,DQ,DR,FILE,FLN,HEDI,ID,IEN  NEW PEDI,SEG,STAT,HCT,NEWID,TSSN,REQSUB,NAFLG,NPFLG,TRUSTED  NEW IBCNACT,IBCNADT,FSVDY,PSVDY  NEW BPSIEN,CMIEN,DATA,DATAAP,DATABPS,DATACM,DATE,ERROR,FIELDNO,FILENO  NEW IBSEG,MSG,BUFF  NEW X12TABLE,BADFMT  ;  ; BADFMT is true if a site with patch 300 receives an eIV message in the previous HL7 interface structure (pre-300)  ;  ; Build local table of file numbers to determine if response is eIV or ePHARM  F D=11:1:18 S X12TABLE("365.0"\_D)=""  F D=21:1:28 S X12TABLE("365.0"\_D)=""  S X12TABLE(350.9)=""      ; IB\*2.0\*506  ;  ; Decide if message belongs to "E-Pharm" or "eIV"  S APP=""  S HCT=0,ERFLG=0  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D SPAR^IBCNEHLU I $G(IBSEG(1))="MFI" S FILE=$G(IBSEG(2)),FLN=$P(FILE,$E(HLECH,1),1) Q  I ",366.01,366.02,366.03,365.12,355.3,"[(","\_FLN\_",") S APP="E-PHARM"  I FLN=365.12 D  . S HCT=0,BADFMT=0  . F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:(APP="IIV")!BADFMT  .. D SPAR^IBCNEHLU  .. I $G(IBSEG(1))="MFE",$P($G(IBSEG(5)),$E(HLECH,1),3)'="" D  Q  ... S BADFMT=1,APP=""  ... S MSG(1)="Log a Remedy Ticket for this issue."  ... S MSG(2)="Please include in the Remedy Ticket that the eIV payer tables may be out"  ... S MSG(3)="of sync with the master list and will need a new copy of the payer table"  ... S MSG(4)="from Austin."  ... D MSG^IBCNEUT5($$MGRP^IBCNEUT5(),"eIV payer tables may be out of synch with master list","MSG(")  .. I $G(IBSEG(1))="ZPA" S APP="IIV"  I $D(X12TABLE(FLN)) S APP="IIV"  ; If neither eIV or ePHARM then quit  I APP="" Q  ;  S HCT=1,NAFLG=0,NPFLG=0,D=""  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:ERFLG  . D SPAR^IBCNEHLU  . S SEG=$G(IBSEG(1))  . ;  . I APP="E-PHARM" D  .. I SEG="MFI" D  ... S FILE=$G(IBSEG(2))  ... S FLN=$P(FILE,$E(HLECH,1),1)  ... ;  ... ; Initialize MFK Message (Application Acknowledgement) variables  ... ; Master File Identifier  ... S DATAMFK("MFI-1")=$G(IBSEG(2))  ... ;  ... ; File-Level Event Code  ... S DATAMFK("MFI-3")=$G(IBSEG(4))  .. ;  .. I SEG="MFE" D  ... I $G(FLN)="" S ERFLG=1,MSG(1)="File Number not found in MFN message" Q  ... I '$$VFILE^DILFD(FLN) S ERFLG=1,MSG(1)="File "\_FLN\_" not found in the Data Dictionary" Q  ... ;  ... ; Initialize MFK Message (Application Acknowledgement) variables  ... ; Record-Level Event Code  ... S DATAMFK("MFE-1")=$G(IBSEG(2))  ... ;  ... ; Primary Key Value  ... S DATAMFK("MFE-4")=$G(IBSEG(5))  ... ;  ... ; Primary Key Value Type  ... S DATAMFK("MFE-5")=$G(IBSEG(6))  ... ;  ... ; Transfer control to e-Pharmacy  ... D ^IBCNRHLT Q  .. ;  .. ; Transfer control on other segments  .. I ",ZCM,ZP0,ZPB,ZPL,ZPT,ZRX,"[(","\_SEG\_",") D ^IBCNRHLT  . ;  . ;  . I APP="IIV" D  .. I SEG="MFI" D  ... S FILE=$G(IBSEG(2))  ... S FLN=$P(FILE,$E(HLECH,1),1)  .. ;  .. I SEG="MFE" D  ... I $G(FLN)="" S ERFLG=1,MSG(1)="File Number not found in MFN message" Q  ... I '$$VFILE^DILFD(FLN) S ERFLG=1,MSG(1)="File "\_FLN\_" not found in the Data Dictionary" Q  ... ;  ... I FLN'=365.12 D  Q  .... S DATA=$G(IBSEG(5))  .... S ID=$$DECHL7^IBCNEHL2($P(DATA,$E(HLECH,1),1)),DESC=$$DECHL7^IBCNEHL2($P(DATA,$E(HLECH,1),2))  .... D TFIL  ... ;  ... ; Pull the action code  ... S IBCNACT=$G(IBSEG(2))  ... ; Effective Date  ... S IBCNADT=$G(IBSEG(4))  .. ;  .. I SEG="ZP0" D  ... S ID=$$DECHL7^IBCNEHL2(IBSEG(3)),NEWID=$$DECHL7^IBCNEHL2(IBSEG(4))  ... S DESC=$$DECHL7^IBCNEHL2(IBSEG(5)),HEDI=$$DECHL7^IBCNEHL2(IBSEG(6)),PEDI=$$DECHL7^IBCNEHL2(IBSEG(7))  .. ;  .. I SEG="ZPA" D  ... S STAT=$S(IBSEG(4)="Y":1,1:0)  ... S TSSN=IBSEG(5),REQSUB=IBSEG(7)  ... S FSVDY=IBSEG(8),PSVDY=IBSEG(9)  ... S TRUSTED=$S(IBSEG(10)="N":0,1:1)  ... D PFIL  Q  ; PFIL ; Payer Table Filer  ; Set the action:  ; MAD=Add, MUP=Update, MDC=Deactivate, MAC=Reactivate  N OLDAF,OLDTF  S IBCNADT=$$FMDATE^HLFNC(IBCNADT)  I IBCNADT="" S IBCNADT=$$NOW^XLFDT()  ; If the action is MAD - Add the payer as new  N IBNOK,IBAPP,IBID,IBDESC,IBSTR  S IBNOK=0,IBAPP=($TR(APP," ")="")  I IBCNACT="MAD" D  I IBNOK G PFILX  . ; Check certain required fields: Application, VA National & Payer Name  . ; If not populated, send MailMan message.  . S IBID=($TR(ID," ")=""),IBDESC=($TR(DESC," ")="")  . S IBNOK=IBAPP!IBID!IBDESC  . I 'IBNOK D MAD(DESC) Q  . S IBSTR="" I IBAPP S IBSTR="Application"  . I IBID S:IBSTR]"" IBSTR=IBSTR\_", " S IBSTR=IBSTR\_"VA National"  . I IBDESC S:IBSTR]"" IBSTR=IBSTR\_", " S IBSTR=IBSTR\_"Payer Name"  . S MSG(1)="MAD action received. "\_IBSTR\_" unknown."  I IBCNACT'="MAD" D FND  N IBCNTYPE  I IEN<1!IBAPP D  G PFILX  . S IBCNTYPE=$S(IBCNACT="MAD":"Add",IBCNACT="MUP":"Update",IBCNACT="MDC":"Deactivate",IBCNACT="MAC":"Reactivate",1:"Unknown")  . S MSG(1)=IBCNTYPE\_" ("\_IBCNACT\_") action received. Payer and/or Application may be unknown."  . S MSG(2)=""  . S MSG(3)="VA National : "\_ID  . S MSG(4)="Payer Name : "\_DESC  . S MSG(5)="Application : "\_APP  . S MSG(6)=""  . S MSG(7)="Log a Remedy Ticket for this issue."  . S MSG(8)=""  . S MSG(9)="Please include in the Remedy Ticket that VISTA did not receive the required"  . S MSG(10)="information or the accurate information to add/update this Payer."  . D MSG^IBCNEUT5($$MGRP^IBCNEUT5(),"eIV payer tables may be out of synch with master list","MSG(")  ;  S DESC=$E(DESC,1,80) ;restriction of the field in the DD  S DIC=$$ROOT^DILFD(FLN)  S DR=".01///^S X=DESC;.02////^S X=NEWID;.05////^S X=PEDI;.06////^S X=HEDI"  ;  ; If new payer, add the Date/Time created  I NPFLG S DR=DR\_";.04///^S X=$$NOW^XLFDT()"  S DIE=DIC,DA=IEN D ^DIE  ;  ; Check for application  S DIC="^IBE(365.13,",DIC(0)="X",X=APP D ^DIC  S AIEN=+Y I AIEN<1 D  . S DLAYGO=365.13,DIC(0)="L",DIC("P")=DLAYGO  . S DIE=DIC,X=APP  . K DD,DO  . D FILE^DICN  . K DO  . S AIEN=+Y  ;  S APIEN=$O(^IBE(365.12,IEN,1,"B",AIEN,""))  I APIEN="" D  . S DLAYGO=365.121,DIC(0)="L",DIC("P")=DLAYGO,DA(1)=IEN,X=AIEN  . S DIC="^IBE(365.12,"\_DA(1)\_",1,",DIE=DIC  . I '$D(^IBE(365.12,IEN,1,0)) S ^IBE(365.12,IEN,1,0)="^365.121P^^"  . K DD,DO  . D FILE^DICN  . K DO  . S APIEN=+Y,NAFLG=1  ; get current values for Active and Trusted flags  S OLDAF=$P(^IBE(365.12,IEN,1,APIEN,0),U,2),OLDTF=$P(^IBE(365.12,IEN,1,APIEN,0),U,7)  S DA(1)=IEN,DA=APIEN,DIC="^IBE(365.12,"\_DA(1)\_",1,",DR=""  ;  I IBCNACT="MDC" S DR=DR\_".11///^S X=1;.12////^S X=IBCNADT;",STAT=0  I IBCNACT="MAC" S DR=DR\_".11///^S X=0;.12///@;"  S DR=DR\_".02///^S X=STAT;.06///^S X=$$NOW^XLFDT();.07///^S X=TRUSTED"  I IBCNACT'="MDC" S DR=DR\_";.08///^S X=REQSUB;.1///^S X=TSSN;.14///^S X=FSVDY;.15///^S X=PSVDY"  ;  ; If new application, add the Date/Time created  I NAFLG S DR=DR\_";.13///^S X=$$NOW^XLFDT()"  ;  S DIE=DIC D ^DIE  ; Update flag logs  I STAT'=OLDAF D UPDLOG("A",STAT,IEN,APIEN)  I TRUSTED'=OLDTF D UPDLOG("T",TRUSTED,IEN,APIEN)  I IBCNACT="MDC" D MDC Q PFILX ;  Q  ; TFIL ; Non Payer Tables Filer  NEW DIC,X,DLAYGO,Y,IEN,MAX  ;  ; store the FILENAME, FIELDNAME and VALUE if the APP is IIV and FLN is 350.9. - IB\*2.0\*506  ; For file #350.9, DESC represents the FIELD NUMBER and ID represents the VALUE.  I APP="IIV",FLN=350.9 D  Q  . S DIE=FLN,DA=1,DR=DESC\_"///"\_ID  . D ^DIE  ;  S DIC(0)="X",X=ID,DIC=$$ROOT^DILFD(FLN)  D ^DIC S IEN=+Y  ; don't update existing entries  I IEN>0 Q  ;  D FIELD^DID(FLN,.02,,"FIELD LENGTH","MAX")  I MAX("FIELD LENGTH")>0 S DESC=$E(DESC,1,MAX("FIELD LENGTH")) ; restriction of the field in the DD  ; add new entry to the table  ;S DLAYGO=FLN,DIC(0)="L",DIC("DR")=".02///"\_DESC  S DLAYGO=FLN,DIC(0)="L",DIC("DR")=".02///^S X=DESC"  K DD,DO D FILE^DICN K DO  Q  ; MAD(X) ; Add an entry  D FND  I IEN>0 G MADX  NEW DIC,DIE,DA,DLAYGO,Y,DR  S DIC=$$ROOT^DILFD(FLN)  S DLAYGO=FLN,DIC(0)="L",DIC("P")=DLAYGO,DIE=DIC  K DD,DO  D FILE^DICN  K DO  S IEN=+Y,NPFLG=1 MADX ;  Q  ; FND ; Find an existing Payer entry  NEW DIC,DIE,X,DA,DLAYGO,Y,DR  S X=ID,DIC(0)="X",D="C",DIC=$$ROOT^DILFD(FLN)  ;  ; Do a lookup with the "C" cross-reference  D IX^DIC  S IEN=+Y  Q  ; MDC ; Check for active transmissions and cancel  NEW STA,HIEN,RIEN,TQIEN  F STA=1,2,4,6 S TQIEN="" D  . F  S TQIEN=$O(^IBCN(365.1,"AC",STA,TQIEN)) Q:TQIEN=""  D  .. ;  .. ; If the record doesn't match the payer, quit  .. I $P(^IBCN(365.1,TQIEN,0),U,3)'=IEN Q  .. ;  .. ; Set the status to 'Cancelled'  .. D SST^IBCNEUT2(TQIEN,7)  .. ;  .. ; If a buffer entry, set to ! (bang)  .. S BUFF=$P(^IBCN(365.1,TQIEN,0),U,5)  .. I BUFF'="" D BUFF^IBCNEUT2(BUFF,17)  .. ;  .. ; Change any responses status also  .. S HIEN=0 F  S HIEN=$O(^IBCN(365.1,TQIEN,2,HIEN)) Q:'HIEN  D  ... S RIEN=$P(^IBCN(365.1,TQIEN,2,HIEN,0),U,3)  ... ; If the Response status is 'Response Received', don't change it  ... I $P(^IBCN(365,RIEN,0),U,6)=3 Q  ... D RSP^IBCNEUT2(RIEN,7)  Q  ; UPDLOG(FLAG,VALUE,PIEN,APIEN) ; Update active/trusted flag logs  ; FLAG - "A" for Active flag, "T" for Trusted flag  ; VALUE - new flag value (0 or 1)  ; PIEN - ien in PAYER file (365.12)  ; APIEN - ien in APPLICATION sub-file (365.121)  ;  N FILE,IENSTR,UPDT  I $G(FLAG)=""!($G(VALUE)="") Q  I +$G(PIEN)=0!(+$G(APIEN)=0) Q  S FILE=$S(FLAG="A":"365.1212",FLAG="T":"365.1213",1:"") I FILE="" Q  S IENSTR="+1,"\_APIEN\_","\_PIEN\_","  S UPDT(FILE,IENSTR,.01)=$$NOW^XLFDT()  S UPDT(FILE,IENSTR,.02)=VALUE  D UPDATE^DIE("E","UPDT")  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNEHLT ;DAOU/ALA - HL7 Process Incoming MFN Messages ; 09 Dec 2005 3:30 PM  ;;2.0;INTEGRATED BILLING;\*\*184,251,271,300,416,438,506\*\*;21-MAR-94;Build 74  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ;\*\*Program Description\*\*  ; This program will process incoming MFN messages and  ; update the appropriate tables  ;  **; References to ePharmacy and ePharmacy processing will be obsolete upon national release of IB\*2\*XXX**  **; A subsequent ePharmacy patch will remove all references to ePharmacy and its processing.**  **; All areas or lines of code that will need to be removed or adjusted as a result of this ePharmacy logic going away**  **; will be indicated below with a reference of “future ePharmacy patch”**  ; EN ; Entry Point  NEW AIEN,APIEN,APP,D0,D,DESC,DQ,DR,FILE,FLN,HEDI,ID,IEN  NEW PEDI,SEG,STAT,HCT,NEWID,TSSN,REQSUB,NAFLG,NPFLG,TRUSTED  NEW IBCNACT,IBCNADT,FSVDY,PSVDY  NEW BPSIEN,CMIEN,DATA,DATAAP,DATABPS,DATACM,DATE,ERROR,FIELDNO,FILENO  NEW IBSEG,MSG,BUFF  NEW X12TABLE,BADFMT  ;  ; BADFMT is true if a site with patch 300 receives an eIV message in the previous HL7 interface structure (pre-300)  ;  ; Build local table of file numbers **~~to determine if response is eIV or ePHARM~~** **; removed with “future ePharmacy patch”**  **; \* Warning: Before adding a new table to be updated by FSC, one must get FSC to agree and the eIV ICD documentation has to be updated and approved by the VA HL7 team. Just adding a table number here does absolutely nothing without involving the other teams.**  F D=11:1:18 S X12TABLE("365.0"\_D)="" **~~F D=21:1:28 S X12TABLE("365.0"\_D)=""~~ ; remove this line as FSC never updated the tables 365.022 thru 365.028**  **S X12TABLE(350.021)=””**   S X12TABLE(350.9)=""      ; IB\*2.0\*506  ;  ; Decide if message belongs to "E-Pharm" or "eIV" **; adjust with “future ePharmacy patch”**  S APP=""  S HCT=0,ERFLG=0  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D SPAR^IBCNEHLU I $G(IBSEG(1))="MFI" S FILE=$G(IBSEG(2)),FLN=$P(FILE,$E(HLECH,1),1) Q  I ",366.01,366.02,366.03,365.12,355.3,"[(","\_FLN\_",") S APP="E-PHARM"  I FLN=365.12 D  . S HCT=0,BADFMT=0  . F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:(APP="IIV")!BADFMT  .. D SPAR^IBCNEHLU  .. I $G(IBSEG(1))="MFE",$P($G(IBSEG(5)),$E(HLECH,1),3)'="" D  Q  ... S BADFMT=1,APP=""  ... S MSG(1)="Log a Remedy Ticket for this issue."  ... S MSG(2)="Please include in the Remedy Ticket that the VistA eIV payer tables may be out"  ... S MSG(3)="of sync with the master list and will need a new copy of the payer table"  ... S MSG(4)="update message from Austin."  ... D MSG^IBCNEUT5($$MGRP^IBCNEUT5(),"eIV payer tables may be out of synch with master list","MSG(")  .. I $G(IBSEG(1))="ZPA" S APP="IIV"  I $D(X12TABLE(FLN)) S APP="IIV" **~~; If neither eIV or ePHARM then quit~~ ; Remove comment with ePharmacy patch**  I APP="" Q  ;  S HCT=1,NAFLG=0,NPFLG=0,D=""  F  S HCT=$O(^TMP($J,"IBCNEHLI",HCT)) Q:HCT=""  D  Q:ERFLG  . D SPAR^IBCNEHLU  . S SEG=$G(IBSEG(1))  . ; **~~. I APP="E-PHARM" D~~** ;remove with “future ePharmacy patch” **~~.. I SEG="MFI" D  ... S FILE=$G(IBSEG(2))  ... S FLN=$P(FILE,$E(HLECH,1),1)  ... ;  ... ; Initialize MFK Message (Application Acknowledgement) variables  ... ; Master File Identifier  ... S DATAMFK("MFI-1")=$G(IBSEG(2))  ... ;  ... ; File-Level Event Code  ... S DATAMFK("MFI-3")=$G(IBSEG(4))  .. ;  .. I SEG="MFE" D  ... I $G(FLN)="" S ERFLG=1,MSG(1)="File Number not found in MFN message" Q  ... I '$$VFILE^DILFD(FLN) S ERFLG=1,MSG(1)="File "\_FLN\_" not found in the Data Dictionary" Q  ... ;  ... ; Initialize MFK Message (Application Acknowledgement) variables  ... ; Record-Level Event Code  ... S DATAMFK("MFE-1")=$G(IBSEG(2))  ... ;  ... ; Primary Key Value  ... S DATAMFK("MFE-4")=$G(IBSEG(5))  ... ;  ... ; Primary Key Value Type  ... S DATAMFK("MFE-5")=$G(IBSEG(6))  ... ;  ... ; Transfer control to e-Pharmacy  ... D ^IBCNRHLT Q  .. ;  .. ; Transfer control on other segments  .. I ",ZCM,ZP0,ZPB,ZPL,ZPT,ZRX,"[(","\_SEG\_",") D ^IBCNRHLT~~**  . ;  . ;  . I APP="IIV" D  .. I SEG="MFI" D  ... S FILE=$G(IBSEG(2))  ... S FLN=$P(FILE,$E(HLECH,1),1)  .. ;  .. I SEG="MFE" D  ... I $G(FLN)="" S ERFLG=1,MSG(1)="File Number not found in MFN message" Q  ... I '$$VFILE^DILFD(FLN) S ERFLG=1,MSG(1)="File "\_FLN\_" not found in the Data Dictionary" Q  ... ;  ... I FLN'=365.12 D  Q  .... S DATA=$G(IBSEG(5))  .... S ID=$$DECHL7^IBCNEHL2($P(DATA,$E(HLECH,1),1)),DESC=$$DECHL7^IBCNEHL2($P(DATA,$E(HLECH,1),2))  .... D TFIL  ... ;  ... ; Pull the action code  ... S IBCNACT=$G(IBSEG(2))  ... ; Effective Date  ... S IBCNADT=$G(IBSEG(4))  .. ;  .. I SEG="ZP0" D  ... S ID=$$DECHL7^IBCNEHL2(IBSEG(3)),NEWID=$$DECHL7^IBCNEHL2(IBSEG(4))  ... S DESC=$$DECHL7^IBCNEHL2(IBSEG(5)),HEDI=$$DECHL7^IBCNEHL2(IBSEG(6)),PEDI=$$DECHL7^IBCNEHL2(IBSEG(7))  .. ;  .. I SEG="ZPA" D  ... S STAT=$S(IBSEG(4)="Y":1,1:0)  ... S TSSN=IBSEG(5),REQSUB=IBSEG(7)  ... S FSVDY=IBSEG(8),PSVDY=IBSEG(9)  ... S TRUSTED=$S(IBSEG(10)="N":0,1:1)  ... D PFIL  Q  ; PFIL ; Payer Table Filer  ; Set the action:  ; MAD=Add, MUP=Update, MDC=Deactivate, MAC=Reactivate  N OLDAF,OLDTF  S IBCNADT=$$FMDATE^HLFNC(IBCNADT)  I IBCNADT="" S IBCNADT=$$NOW^XLFDT()  ; If the action is MAD - Add the payer as new  N IBNOK,IBAPP,IBID,IBDESC,IBSTR  S IBNOK=0,IBAPP=($TR(APP," ")="")  I IBCNACT="MAD" D  I IBNOK G PFILX  . ; Check certain required fields: Application, VA National & Payer Name  . ; If not populated, send MailMan message.  . S IBID=($TR(ID," ")=""),IBDESC=($TR(DESC," ")="")  . S IBNOK=IBAPP!IBID!IBDESC  . I 'IBNOK D MAD(DESC) Q  . S IBSTR="" I IBAPP S IBSTR="Application"  . I IBID S:IBSTR]"" IBSTR=IBSTR\_", " S IBSTR=IBSTR\_"VA National"  . I IBDESC S:IBSTR]"" IBSTR=IBSTR\_", " S IBSTR=IBSTR\_"Payer Name"  . S MSG(1)="MAD action received. "\_IBSTR\_" unknown."  I IBCNACT'="MAD" D FND  N IBCNTYPE  I IEN<1!IBAPP D  G PFILX  . S IBCNTYPE=$S(IBCNACT="MAD":"Add",IBCNACT="MUP":"Update",IBCNACT="MDC":"Deactivate",IBCNACT="MAC":"Reactivate",1:"Unknown")  . S MSG(1)=IBCNTYPE\_" ("\_IBCNACT\_") action received. Payer and/or Application may be unknown."  . S MSG(2)=""  . S MSG(3)="VA National : "\_ID  . S MSG(4)="Payer Name : "\_DESC  . S MSG(5)="Application : "\_APP  . S MSG(6)=""  . S MSG(7)="Log a Remedy Ticket for this issue."  . S MSG(8)=""  . S MSG(9)="Please include in the Remedy Ticket that VISTA did not receive the required"  . S MSG(10)="information or the accurate information to add/update this Payer."  . D MSG^IBCNEUT5($$MGRP^IBCNEUT5(),"eIV payer tables may be out of synch with master list","MSG(")  ;  S DESC=$E(DESC,1,80) ;restriction of the field in the DD  S DIC=$$ROOT^DILFD(FLN)  S DR=".01///^S X=DESC;.02////^S X=NEWID;.05////^S X=PEDI;.06////^S X=HEDI"  ;  ; If new payer, add the Date/Time created  I NPFLG S DR=DR\_";.04///^S X=$$NOW^XLFDT()"  S DIE=DIC,DA=IEN D ^DIE  ;  ; Check for application  S DIC="^IBE(365.13,",DIC(0)="X",X=APP D ^DIC  S AIEN=+Y I AIEN<1 D  . S DLAYGO=365.13,DIC(0)="L",DIC("P")=DLAYGO  . S DIE=DIC,X=APP  . K DD,DO  . D FILE^DICN  . K DO  . S AIEN=+Y  ;  S APIEN=$O(^IBE(365.12,IEN,1,"B",AIEN,""))  I APIEN="" D  . S DLAYGO=365.121,DIC(0)="L",DIC("P")=DLAYGO,DA(1)=IEN,X=AIEN  . S DIC="^IBE(365.12,"\_DA(1)\_",1,",DIE=DIC  . I '$D(^IBE(365.12,IEN,1,0)) S ^IBE(365.12,IEN,1,0)="^365.121P^^"  . K DD,DO  . D FILE^DICN  . K DO  . S APIEN=+Y,NAFLG=1  ; get current values for Active and Trusted flags  S OLDAF=$P(^IBE(365.12,IEN,1,APIEN,0),U,2),OLDTF=$P(^IBE(365.12,IEN,1,APIEN,0),U,7)  S DA(1)=IEN,DA=APIEN,DIC="^IBE(365.12,"\_DA(1)\_",1,",DR=""  ;  I IBCNACT="MDC" S DR=DR\_".11///^S X=1;.12////^S X=IBCNADT;",STAT=0  I IBCNACT="MAC" S DR=DR\_".11///^S X=0;.12///@;"  S DR=DR\_".02///^S X=STAT;.06///^S X=$$NOW^XLFDT();.07///^S X=TRUSTED"  I IBCNACT'="MDC" S DR=DR\_";.08///^S X=REQSUB;.1///^S X=TSSN;.14///^S X=FSVDY;.15///^S X=PSVDY"  ;  ; If new application, add the Date/Time created  I NAFLG S DR=DR\_";.13///^S X=$$NOW^XLFDT()"  ;  S DIE=DIC D ^DIE  ; Update flag logs  I STAT'=OLDAF D UPDLOG("A",STAT,IEN,APIEN)  I TRUSTED'=OLDTF D UPDLOG("T",TRUSTED,IEN,APIEN)  I IBCNACT="MDC" D MDC Q PFILX ;  Q  ; TFIL ; Non Payer Tables Filer **~~NEW DIC,X,DLAYGO,Y,IEN,MAX~~**  **N DESC,DIC,DLAYGO,EXTRACT,X,Y,IEN,MAX ; 2.6.10.17 Added DESC and EXTRACT**   ;  ; store the FILENAME, FIELDNAME and VALUE if the APP is IIV and FLN is 350.9. - IB\*2.0\*506  ; For file #350.9, DESC represents the FIELD NUMBER and ID represents the VALUE.  I APP="IIV",FLN=350.9 D  Q  . S DIE=FLN,DA=1,DR=DESC\_"///"\_ID  . D ^DIE  ;  **; 2.6.10.17 Added**  **I APP="IIV",FLN=350.9002 D  Q  . S EXTRACT=$E(DESC,1,5) ; Either “Batch” or “Appt “  . S DESC=$E(DESC,6,99) ; Field number  . ;  . ; Look up what DA should be based on value of EXTRACT and set DA and file  . ; should look something like below   . S DIE=FLN,DR=DESC\_"///"\_ID   . D ^DIE**  ;  S DIC(0)="X",X=ID,DIC=$$ROOT^DILFD(FLN)  D ^DIC S IEN=+Y  ; don't update existing entries  I IEN>0 Q  ;  D FIELD^DID(FLN,.02,,"FIELD LENGTH","MAX")  I MAX("FIELD LENGTH")>0 S DESC=$E(DESC,1,MAX("FIELD LENGTH")) ; restriction of the field in the DD  ; add new entry to the table  ;S DLAYGO=FLN,DIC(0)="L",DIC("DR")=".02///"\_DESC  S DLAYGO=FLN,DIC(0)="L",DIC("DR")=".02///^S X=DESC"  K DD,DO D FILE^DICN K DO  Q  ; MAD(X) ; Add an entry  D FND  I IEN>0 G MADX  NEW DIC,DIE,DA,DLAYGO,Y,DR  S DIC=$$ROOT^DILFD(FLN)  S DLAYGO=FLN,DIC(0)="L",DIC("P")=DLAYGO,DIE=DIC  K DD,DO  D FILE^DICN  K DO  S IEN=+Y,NPFLG=1 MADX ;  Q  ; FND ; Find an existing Payer entry  NEW DIC,DIE,X,DA,DLAYGO,Y,DR  S X=ID,DIC(0)="X",D="C",DIC=$$ROOT^DILFD(FLN)  ;  ; Do a lookup with the "C" cross-reference  D IX^DIC  S IEN=+Y  Q  ; MDC ; Check for active transmissions and cancel  NEW STA,HIEN,RIEN,TQIEN  F STA=1,2,4,6 S TQIEN="" D  . F  S TQIEN=$O(^IBCN(365.1,"AC",STA,TQIEN)) Q:TQIEN=""  D  .. ;  .. ; If the record doesn't match the payer, quit  .. I $P(^IBCN(365.1,TQIEN,0),U,3)'=IEN Q  .. ;  .. ; Set the status to 'Cancelled'  .. D SST^IBCNEUT2(TQIEN,7)  .. ;  .. ; If a buffer entry, set to ! (bang)  .. S BUFF=$P(^IBCN(365.1,TQIEN,0),U,5)  .. I BUFF'="" D BUFF^IBCNEUT2(BUFF,17)  .. ;  .. ; Change any responses status also  .. S HIEN=0 F  S HIEN=$O(^IBCN(365.1,TQIEN,2,HIEN)) Q:'HIEN  D  ... S RIEN=$P(^IBCN(365.1,TQIEN,2,HIEN,0),U,3)  ... ; If the Response status is 'Response Received', don't change it  ... I $P(^IBCN(365,RIEN,0),U,6)=3 Q  ... D RSP^IBCNEUT2(RIEN,7)  Q  ; UPDLOG(FLAG,VALUE,PIEN,APIEN) ; Update active/trusted flag logs  ; FLAG - "A" for Active flag, "T" for Trusted flag  ; VALUE - new flag value (0 or 1)  ; PIEN - ien in PAYER file (365.12)  ; APIEN - ien in APPLICATION sub-file (365.121)  ;  N FILE,IENSTR,UPDT  I $G(FLAG)=""!($G(VALUE)="") Q  I +$G(PIEN)=0!(+$G(APIEN)=0) Q  S FILE=$S(FLAG="A":"365.1212",FLAG="T":"365.1213",1:"") I FILE="" Q  S IENSTR="+1,"\_APIEN\_","\_PIEN\_","  S UPDT(FILE,IENSTR,.01)=$$NOW^XLFDT()  S UPDT(FILE,IENSTR,.02)=VALUE  D UPDATE^DIE("E","UPDT")  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNEDE | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.10.22 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | EN^IBCNEDE1  EN^IBCNEDE2  IBCNEDEP  IBCNEHLM  MMPURGE^IBCNEKI2  PAYR^IBCNEUT2  MGRP^IBCNEUT5  PYRAPP^IBCNEUT5  MSG^IBCNEUT5 AMCHECK^IBCNEUT6 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  .  EN ; Entry Point  ; Prevent simultaneous runs  ; Set error trap to ensure that lock is released  N $ES,$ET  S $ET="D ER^IBCNEDE"  ; Check lock  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  .  EN ; Entry Point  ; Prevent simultaneous runs  ; Set error trap to ensure that lock is released  N $ES,$ET **;**  **; 2.6.10.22 Quit if Nightly Extract Master switch is off**  **Q:$$GET1^DIQ(350.9,"1,",50.28,"I")**   S $ET="D ER^IBCNEDE" .  . | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNERTQ | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.10.18 | | | | | | | | |
| **Related Options** | eIV NIGHTLY PROCESS [IBCNE IIV BATCH PROCESS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBU1  IBCNEDE1 | | | | | GENERATE^HLMA  SYMBOL^IBCNBLL  SETREL^IBCNEDE1  TQ^IBCNEDE1  UPDDTS^IBCNEDE6  PYRACTV^IBCNEDE7  SETTINGS^IBCNEDE7  PROC^IBCNEDEP  HLER^IBCNEDEQ  SCC^IBCNEDEQ  INIT^IBCNEHLO  BUFF^IBCNEUT2  INSERROR^IBCNEUT3  CLEAR^IBCNEUT4  MGRP^IBCNEUT5  TQUPDSV^IBCNEUT5 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| .  .  IBE(IEN) ; Insurance Buffer Extract  N SETSTR,FRESHDAY,ISYMBOL,STATIEN,OVRFRESH  N PDOD,SRVICEDT,FRESHDT,PAYERSTR,PAYERID,SYMBOL  N PIEN,INSNAME,MCAREFLG,TQDT,TQIENS,TQOK,QUEUED  N TQIEN,CNT,SIDCNT,MAXCNT  ;  S CNT=0  ;  S QUEUED=0  S SETSTR=$$SETTINGS^IBCNEDE7(1) ;Returns buffer extract settings  I 'SETSTR Q QUEUED                  ;Quit if extract is not active  S MAXCNT=$P(SETSTR,U,4) ;Max # TQ entries that may be created  S:MAXCNT="" MAXCNT=9999999999  ;  S FRESHDAY=$P($G(^IBE(350.9,1,51)),U,1) ;System freshness days  ;  ; Get symbol, if symbol'=" " OR "!" OR "#" then quit  S ISYMBOL=$$SYMBOL^IBCNBLL(IEN) ;Insurance buffer symbol  I (ISYMBOL'=" ")&(ISYMBOL'="!")&(ISYMBOL'="#") Q QUEUED.   ;  ; Get the eIV STATUS IEN and quit for response related errors  S STATIEN=+$P($G(^IBA(355.33,IEN,0)),U,12)  .  . | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| .  .  IBE(IEN) ; Insurance Buffer Extract  N SETSTR,FRESHDAY,ISYMBOL,STATIEN,OVRFRESH  N PDOD,SRVICEDT,FRESHDT,PAYERSTR,PAYERID,SYMBOL  N PIEN,INSNAME,MCAREFLG,TQDT,TQIENS,TQOK,QUEUED  N TQIEN,CNT,SIDCNT,MAXCNT  ;  S CNT=0  ;  S QUEUED=0  S SETSTR=$$SETTINGS^IBCNEDE7(1) ;Returns buffer extract settings  I 'SETSTR Q QUEUED                  ;Quit if extract is not active  S MAXCNT=$P(SETSTR,U,4) ;Max # TQ entries that may be created  S:MAXCNT="" MAXCNT=9999999999  ;  S FRESHDAY=$P($G(^IBE(350.9,1,51)),U,1) ;System freshness days  ;  ; Get symbol, if symbol'=" " OR "!" OR "#" then quit  S ISYMBOL=$$SYMBOL^IBCNBLL(IEN) ;Insurance buffer symbol  I (ISYMBOL'=" ")&(ISYMBOL'="!")&(ISYMBOL'="#") Q QUEUED.  **;**  **; 2.6.10.18 Quit if Realtime Extract Master switch is off**  **; Note: Checking here instead of the top of TRIG to check for above error conditions first**  **Q:$$GET1^DIQ(350.9,"1,",50.27,"I") 0**   ;  ; Get the eIV STATUS IEN and quit for response related errors  S STATIEN=+$P($G(^IBA(355.33,IEN,0)),U,12)  .  . | | | | | | | | | |

##### Templates (Entry Points)

| **Template Name** | | **IBJP IIV SITE PARAMETERS** | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Enhancement Category** | | New | Modify | Delete | | No Change |
| **RSD** | | 2.6.10.19 | | | | |
| **Template Type** | | Sort | Input | Print | | Other |
| **Related Options** | | VALM HIDDEN ACTIONS | | | | |
| **Related Routines** | | **Routines “Called By”** | | | **Routines “Called”** | |
|  | | IBJPI | | | FO^IBCNEUT1 MGRP^IBCNEUT5 | |
| Routines | Description | | | | |
| **Data Dictionary (DD) References** | None | | | | |
| **Global References** | ^TMP($J,”IBJPI”) | | | | |

| **Input Template** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Input Template Name** | **IBCNE GENERAL PARAMETER EDIT** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.10.2, 2.6.10.9, 2.6.10.11, 2.6.10.13, 2.6.10.14, 2.6.10.15, 2.6.10.16 | | | | | | | | |
| **Related Options** | None | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IIVEDIT^IBJPI2 | | | | | NONE | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | IBJP IIV GENERAL EDIT | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| MESSAGES MAILGROUP  CONTACT PERSON  CONTACT PERSON:  OFFICE PHONE;REQ  EMAIL ADDRESS;REQ  FAILURE MAILMAN MSG | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **51.25;"Medicare Payer"**  **13.01;"HMS Directory"**  **13.02;"EII Active"** | | | | | | | | | |

#### System Feature: Reports

RSD 2.6.11.1 Report (List Plans by Insurance Company) shall filter data based on an insurance company list selection with options of ACTIVE/INACTIVE/ALL

RSD 2.6.11.2 Report (List Plans by Insurance Company) shall provide a list of insurance companies to the user for selection in a ListMan display (only applies if the user selected to view ACTIVE or INACTIVE insurance companies) thus allowing a user to select one, or several insurance companies. (i.e. 1, 3-5, 7)

RSD 2.6.11.3 Report (List Plans by Insurance Company) shall allow the user to filter the group plan list selectionwith options of ACTIVE/INACTIVE/BOTH.

RSD 2.6.11.4 Report (List Plans by Insurance Company) shall display the following additional information regardless if the report is run with/without the subscriber detail - Timely Filing (FTF) at the insurance company level, Electronic Plan Type at the group level, Timely Filing (FTF) at the group level, Type of Plan at the group level.

RSD 2.6.11.5 Report (List Plans by Insurance Company) shall display the Patient ID if the report is run with the subscriber detail included.

RSD 2.6.11.6 Report (List Plans by Insurance Company) shall allow the report to be printed to paper.

RSD 2.6.11.7 Report (List Plans by Insurance Company) shall be modified so that when the report is run with subscriber detail the existing column ‘SUBSCRIBER NAME/ID’ is split into the following two separate columns: ‘SUBSCRIBER NAME’ and ‘SSN’.

RSD 2.6.11.8 Report (List Plans by Insurance Company) shall be modified so that when the report is run with subscriber detail the last 4 digits of the SSN will be displayed without the prefix of the initial of the patient’s last name and without brackets.

RSD 2.6.11.9 Report (List Plans by Insurance Company) shall display an indicator for those groups/insurance companies that are inactive.

RSD 2.6.11.10 Report (List Plans by Insurance Company) shall not display the Phone number for the insurance company.

RSD 2.6.11.11 Report (List Plans by Insurance Company) shall not display the Precert Phone number for the insurance company.

RSD 2.6.11.12 Report (List Plans by Insurance Company) shall not display the ‘ACTIVE/INACTIVE' column.

RSD 2.6.11.13 Report (List Plans by Insurance Company) shall not display the 'GROUP OR IND' column, but will display an indicator for those plans that are individual plans.

RSD 2.6.11.14 Report (List Plans by Insurance Company) shall not display the 'EMPLOYER’ column.

RSD 2.6.11.15 Report (List Plans by Insurance Company) shall not display the 'BEN. USED?' column.

RSD 2.6.11.16 Report (List Plans by Insurance Company) shall not display the column ‘ANN. BEN?’ column.

RSD 2.6.11.17 Report (List Plans by Insurance Company) - Newly added or updated features in the report shall have corresponding help text when user enters a question mark.

RSD 2.6.11.18 Report (List Plans by Insurance Company) - Updated "List Plans by Insurance Company" report shall display an "End of Report" tag regardless if the report was run with/without subscriber detail.

RSD 2.6.12.1 Report (Missing Data) - A new option shall be created to allow a user to run a new "Missing Data" type of report.

RSD 2.6.12.2 Report (Missing Data) – The report shall only include active insurance companies.

RSD 2.6.12.3 Report (Missing Data) – The user shall be prompted for all active insurance companies or the ability to select individual active companies. (Help text included.)

RSD 2.6.12.4 Report (Missing Data) – Shall provide a list of insurance companies to the user for selection in a ListMan display (only applies if the user responded they wanted to select individual insurance companies) thus allowing a user to select one, or several insurance companies. (i.e. 1, 3-5, 7).

RSD 2.6.12.5 Report (Missing Data) – Shall include the ability for a user to apply any or all of the following filters with the default being YES for each of the prompts: active groups with missing Group Number, active groups missing Type of Plan, active groups with missing Timely Filing Time Frame, active groups missing Electronic Plan Type, active groups missing Coverage Limitations, active groups missing BIN, active groups missing PCN. (Includes help text). The filters work independently of each other. In other words, as long as the record with the missing data meets one of the filters then it will fall on the report. It does not have to meet all the filters to fall on the report.

RSD 2.6.12.6 Report (Missing Data) – Shall require the user to select at least one of the missing data filters before the report can be run.

RSD 2.6.12.7 Report (Missing Data) – Shall indicate the fields where data is missing by displaying number signs “#” on the report where the data should be.

RSD 2.6.12.8 Report (Missing Data) – Shall only include records that have missing data elements and meet the user’s selected filters.

RSD 2.6.12.9 Report (Missing Data) - Shall display the "End of Report" tag.

RSD 2.6.12.10 Report (Missing Data) – User shall be able to print the report to paper.

RSD 2.6.13.1 Report (Patients Without MEDICARE) shall add new filtering criteria based on Last Appointment Date using a user defined date range (from/to) with appropriate help text when user enters a question mark.

RSD 2.6.13.2 Report (Patients Without MEDICARE) - The new appointment date range shall honor the following business rules: the user must enter both Start and End dates (they are required), there is no default date for the Start date, the default for the End date is today’s date, the End date cannot be a date in the future.

RSD 2.6.13.3 Report (Patients Without MEDICARE) shall allow a user to pick a single day for the “Last Appointment Date”, meaning that from/to shall be allowed to be the same date.

RSD 2.6.13.4 Report (Patients Without MEDICARE) shall display LAST VERIFIED DATE of the patient policy on the report data output.

RSD 2.6.13.5 Report (Patients Without MEDICARE) shall have the ability to print the report to paper.

RSD 2.6.13.6 Report (Patients Without MEDICARE) shall allow a user the ability to export the data in a delimited file format that can be used by Microsoft Excel.

RSD 2.6.13.7 Report (Patients Without MEDICARE) - The primary sort shall be appointment date in descending (from top to bottom) manner, meaning that the most current appointment date shall be at the top of the report.

RSD 2.6.13.8 Report (Patients Without MEDICARE) – The existing prompt for the sort shall be modified in the following manner: indicate in the help text and instructions on the screen that it is the secondary sort (where appointment date is the primary sort), drop appointment date as an option for the secondary sort, keep the Patient Name as the default for the secondary sort.

RSD 2.6.13.9 Report (Patients Without MEDICARE) - The newly added or updated elements shall have corresponding help text when user enters a question mark.

RSD 2.6.14.1 Report (Active Policies with no Effective Date) shall filter patient policies based on living patients/deceased patients/both with a default of living patients.

RSD 2.6.14.2 Report (Active Policies with no Effective Date) shall filter based on Last Appointment Date with a user defined date range (from/to) regardless of initial sort criteria.

RSD 2.6.14.3 Report (Active Policies with no Effective Date) shall display LAST VERIFIED DATE of the patient policy using a 2 digit year on the report.

RSD 2.6.14.4 Report (Active Policies with no Effective Date) shall display LAST VERIFIED BY of the patient policy listed on the report data output.

RSD 2.6.14.5 Report (Active Policies with no Effective Date) shall allow a user the ability to print the report to paper.

RSD 2.6.14.6 Report (Active Policies with no Effective Date) shall allow a user the ability to export the data in a delimited file format that can be used by Microsoft Excel.

RSD 2.6.14.7 Report (Active Policies with no Effective Date) - The date of death (as a 2 digit year) shall be moved to the last column on the report (far right side of the report).

RSD 2.6.14.8 Report (Active Policies with no Effective Date) - Updated "Active Policies with no Effective Date" reports shall display the "End of Report" tag.

RSD 2.6.14.9 Report (Active Policies with no Effective Date) shall display column headers.

RSD 2.6.14.10 Report (Active Policies with no Effective Date) - SSN shall be trimmed from all to last 4, phone number shall not be displayed, "Reimb VA? Y/N" shall not be displayed, Age shall not be displayed, the label “Whose” and its associated value shall not be displayed, and group name shall not be displayed.

RSD 2.6.14.11 Report (Active Policies with no Effective Date) – The report shall be modified so that the following data elements will be added to the report: group number, Who verified coverage (VC By), Last Appointment date (2 digit year).

RSD 2.6.14.12 Report (Active Policies with no Effective Date) - The newly added or updated prompts shall have a corresponding help text when user enters a question mark.

RSD 2.6.15.1 Report (eIV Auto Update Report) – The existing ‘eIV Patient Insurance Update’ report shall be renamed to ‘eIV Auto Update Report’ while maintaining the existing (IU) shortcut.

RSD 2.6.15.2 Report (eIV Auto Update Report) – The existing prompt ‘Select Insurance Company’ shall be changed to ‘Select Payer’.

RSD 2.6.15.3 Report (eIV Auto Update Report) – When displaying the list of available payers the report shall only include those payers that have the eIV auto update flag defined as yes.

RSD 2.6.15.4 Report (eIV Auto Update Report) – Regardless of the type of report (summary or detail) for each payer selected (unless the user selected all payers) the system shall prompt whether or not the user wants to display associated insurance companies.

RSD 2.6.15.5 Report (eIV Auto Update Report) – If user wants to display insurance companies the system shall prompt the user for insurance companies filtering only those linked to the payer.

RSD 2.6.15.6 Report (eIV Auto Update Report) – The system shall allow the user to select none, one, or multiple insurance companies associated with a given payer.

RSD 2.6.15.7 Report (eIV Auto Update Report) – The report shall only include the ‘autoupdate’ person.

RSD 2.6.15.8 Report (eIV Auto Update Report) – The summary version of the report shall only include the user selected insurance companies associated with a given payer (with a total count of auto updated policies for that insurance company).

RSD 2.6.15.9 Report (eIV Auto Update Report) – The summary version of the report shall include a total count of auto updated policies per selected payer; however, if insurance companies were selected by the user for that payer then the count for that payer is the sum of the auto updated policies for only the selected associated insurance companies.

RSD 2.6.15.10 Report (eIV Auto Update Report) – The report shall be sorted by the Payer’s name.

RSD 2.6.15.11 Report (eIV Auto Update Report) – The detailed version of the report shall not display the following data elements: date the eIV response was received, ‘Ck AB’, the ‘Clerk/Auto’, and ‘Verified’.

RSD 2.6.15.12 Report (eIV Auto Update Report) – The detailed version of the report shall include the following new data elements: Insurance Company, date the eIV inquiry was sent, the date the policy was auto updated, and the eIV Trace number.

RSD 2.6.15.13 Report (eIV Auto Update Report) shall have the ability to print report data output to paper.

RSD 2.6.15.14 Report (eIV Auto Update Report) shall allow a user the ability to export the data in a delimited file format that can be used by Microsoft Excel.

#### Design Element Tables (Reports)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOPP** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.11.1, 2.6.11.2, 2.6.11.3, 2.6.11.6 | | | | | | | | |
| **Related Options** | List Plans by Insurance Company [IBCN LIST PLANS BY INS CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | OK^IBCNSM3  LKP^IBCNSU2  SELI^IBCOPP1  SELP^IBCOPP1  SELR^IBCOPP1  IBCOPP2 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOPP ;ALB/NLR - LIST INS. PLANS BY CO. (DRIVER) ; 08-SEP-94  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28,62\*\*; 21-MAR-94  ; EN ; Describe report   W !!?5,"This report will generate a list of insurance plans by company."  W !?5,"It will help you identify duplicates and verify patient coverage."  W !?5,"You must select one, many (up to 20) or all of the insurance companies;"  W !?5,"anywhere from one to all of the plans under each company; and whether to"  W !?5,"include the patient policies (subscribers) under each plan. The number of"  W !?5,"plans you select is independent for each company you are including, but"  W !?5,"subscriber selection is the same (all or none) for all companies and"  W !?5,"plans within a report. Regardless of how you run the report, the"  W !?5,"number of subscribers per plan will be included.",!!  ;  ; Prompt user to select report type, insurance companies, plans  ;  ; Output from user selections:  ;  ; IBAPA=0 -- list insurance plans by company  ; IBAPA=1 -- list Insurance plans by company with subscriber information  ; IBAI=0 -- user selects insurance companies  ; IBAI=1 -- run report for all insurance companies with plans  ; IBAPL=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies)  ; IBAPL=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ;  S IBAPA=$$SELR^IBCOPP1 I IBAPA<0 G ENQ  S IBAI=$$SELI^IBCOPP1 I IBAI<0 G ENQ  S IBAPL=$$SELP^IBCOPP1 I IBAPL<0 G ENQ  ;  ; obtain plans for selected insurance companies  ;  I IBAI,IBAPL G DEVICE  D START I IBQUIT G ENQ  I '$D(^TMP("IBINC",$J)) W !!,"No plans selected!" G ENQ  ; DEVICE ; Ask user to select device  ;  W !!,"\*\*\* You will need a 132 column printer for this report. \*\*\*",!  S %ZIS="QM" D ^%ZIS G:POP ENQ  I $D(IO("Q")) D  G ENQ  .S ZTRTN="^IBCOPP2",ZTDESC="IB - LIST OF PLANS BY INSURANCE COMPANY"  .F I="^TMP(""IBINC"",$J,","IBAPA","IBAI","IBAPL" S ZTSAVE(I)=""  .D ^%ZTLOAD K IO("Q") D HOME^%ZIS  .W !!,$S($D(ZTSK):"This job has been queued as task #"\_ZTSK\_".",1:"Unable to queue this job.")  .K ZTSK,IO("Q")  ;  ; Compile and print report  ;  U IO D ^IBCOPP2  ; ENQ K DIRUT,DIROUT,DUOUT,DTOUT,IBAPA,IBAI,IBAPL,IBQUIT,X,Y,^TMP("IBINC",$J)  Q  ;  ; START ; Gather plans for all selected companies.  S (IBCT,IBQUIT)=0 K ^TMP("IBINC",$J)  ;  ; - allow user selection of companies if required  I 'IBAI D  I Y<0 S IBQUIT=1 G STARTQ  .S DIC="^DIC(36,",DIC("S")="I $D(^IBA(355.3,""B"",Y))"  .S VAUTSTR="insurance company",VAUTNI=2,VAUTVB="VAUTI",VAUTNALL=1  .D FIRST^VAUTOMA K DIC,VAUTSTR,VAUTNI,VAUTVB,VAUTNALL Q:Y<0  .S IBCNS="" F  S IBCNS=$O(VAUTI(IBCNS)) Q:IBCNS=""  S ^TMP("IBINC",$J,$E(VAUTI(IBCNS),1,25),IBCNS)=""  I IBAPL G STARTQ  ;  ; - gather all companies if required  I IBAI S A=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S ^TMP("IBINC",$J,$E($P($G(^DIC(36,A,0)),"^"),1,25),A)=""  ;  ; - gather plans for selected companies  S IBIC="" F  S IBIC=$O(^TMP("IBINC",$J,IBIC)) Q:IBIC=""!IBQUIT  D  .S IBCNS="" F  S IBCNS=$O(^TMP("IBINC",$J,IBIC,IBCNS)) Q:IBCNS=""!(IBQUIT) D  ..S IBCT=IBCT+1 W !!,"Insurance Company # "\_IBCT\_": "\_IBIC  ..D OK^IBCNSM3 Q:IBQUIT  I 'IBOK K ^TMP("IBINC",$J,IBIC,IBCNS) S IBAI=0 Q  ..W " ...building a list of plans..."  ..K IBSEL,^TMP($J,"IBSEL") D LKP^IBCNSU2(IBCNS,1,1,.IBSEL,0,1) Q:IBQUIT  ..I '$O(^TMP($J,"IBSEL",0)) K ^TMP("IBINC",$J,IBIC,IBCNS) S IBAI=0 Q  ..;  ..; - set plans into an array  ..S IBPN=0 F  S IBPN=$O(^TMP($J,"IBSEL",IBPN)) Q:'IBPN  S ^TMP("IBINC",$J,IBIC,IBCNS,IBPN)=""  ; STARTQ K IBCNS,IBIC,IBJJ,IBCT,IBLCT,IBOK,IBPN,IBSEL,VAUTI,VAUTP,^TMP($J,"IBSEL")  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOPP ;ALB/NLR - LIST INS. PLANS BY CO. (DRIVER) ; 08-SEP-94  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28,62\*\*; 21-MAR-94  ; EN ; Describe report   W !!?5,"This report will generate a list of insurance plans by company."  W !?5,"It will help you identify duplicates **~~and verify patient coverage~~**." ; 2.6.11.1  W !?5,"You must select one, many (up to 20) or all of the insurance companies;"  W !?5,"anywhere from one to all of the plans under each company; and whether to"  W !?5,"include the patient policies (subscribers) under each plan. The number of"  W !?5,"plans you select is independent for each company you are including, but"  W !?5,"subscriber selection is the same (all or none) for all companies and"  W !?5,"plans within a report. Regardless of how you run the report, the" **W !?5,"number of group plans per insurance company and the number of subscribers“**  **W !?5,”per plan will be included.",!! ;2.6.11.1**  ;  ; Prompt user to select report type, insurance companies, plans  ;  ; Output from user selections:  ;  ; IBAPA=0 -- list insurance plans by company  ; IBAPA=1 -- list Insurance plans by company with subscriber information  ; IBAI=0 -- user selects insurance companies  ; IBAI=1 -- run report for all insurance companies with plans  ; IBAPL=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies)  ; IBAPL=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ;  S IBAPA=$$SELR^IBCOPP1 I IBAPA<0 G ENQ  S IBAI=$$SELI^IBCOPP1 I IBAI<0 G ENQ **S IBAPAI=$$SELA^IBCOPP1 I IBAPAI<0 G ENQ ; Select Active/Inactive Both 2.6.11.1**  S IBAPL=$$SELP^IBCOPP1 I IBAPL<0 G ENQ  **S IBAPlAI=$$SELPA^IBCOPP1 I IBAPAI<0 G ENQ ; Select Active/Inactive Both 2.6.11.3**  ; obtain plans for selected insurance companies  ;  I IBAI,IBAPL G DEVICE  D START I IBQUIT G ENQ  I '$D(^TMP("IBINC",$J)) W !!,"No plans selected!" G ENQ  ; DEVICE ; Ask user to select device **;2.6.11.6 – Already allows print to paper**  ;  W !!,"\*\*\* You will need a 132 column printer for this report. \*\*\*",!  S %ZIS="QM" D ^%ZIS G:POP ENQ  I $D(IO("Q")) D  G ENQ  .S ZTRTN="^IBCOPP2",ZTDESC="IB - LIST OF PLANS BY INSURANCE COMPANY"  .F I="^TMP(""IBINC"",$J,","IBAPA","IBAI","IBAPL" S ZTSAVE(I)=""  .D ^%ZTLOAD K IO("Q") D HOME^%ZIS  .W !!,$S($D(ZTSK):"This job has been queued as task #"\_ZTSK\_".",1:"Unable to queue this job.")  .K ZTSK,IO("Q")  ;  ; Compile and print report  ;  U IO D ^IBCOPP2  ; ENQ K DIRUT,DIROUT,DUOUT,DTOUT,IBAPA,IBAI,IBAPL,IBQUIT,X,Y,^TMP("IBINC",$J)  Q  ;  ; START ; Gather plans for all selected companies.   S (IBCT,IBQUIT)=0 K ^TMP("IBINC",$J)  ;  ; - allow user selection of companies if required  **; Replace code below with a call to a new listman template for Insurance selection 2.6.11.2** **~~I 'IBAI D  I Y<0 S IBQUIT=1 G STARTQ  .S DIC="^DIC(36,",DIC("S")="I $D(^IBA(355.3,""B"",Y))"  .S VAUTSTR="insurance company",VAUTNI=2,VAUTVB="VAUTI",VAUTNALL=1  .D FIRST^VAUTOMA K DIC,VAUTSTR,VAUTNI,VAUTVB,VAUTNALL Q:Y<0  .S IBCNS="" F  S IBCNS=$O(VAUTI(IBCN~~**S)) **~~Q:IBCNS=""  S ^TMP("IBINC",$J,$E(VAUTI(IBCNS),1,25),IBCNS)=""~~**  I IBAPL G STARTQ  ;  ; - gather all companies if required  **; 2.6.11.1 only gather companies for Active/Inactive or both based on previous selection below**  **I IBAI S A=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S ^TMP("IBINC",$J,$E($P($G(^DIC(36,A,0)),"^"),1,25),A)=""**  ;  ; - gather plans for selected companies  S IBIC="" F  S IBIC=$O(^TMP("IBINC",$J,IBIC)) Q:IBIC=""!IBQUIT  D  .S IBCNS="" F  S IBCNS=$O(^TMP("IBINC",$J,IBIC,IBCNS)) Q:IBCNS=""!(IBQUIT) D  ..S IBCT=IBCT+1 W !!,"Insurance Company # "\_IBCT\_": "\_IBIC  ..D OK^IBCNSM3 Q:IBQUIT  I 'IBOK K ^TMP("IBINC",$J,IBIC,IBCNS) S IBAI=0 Q  ..W " ...building a list of plans..."  ..K IBSEL,^TMP($J,"IBSEL") D LKP^IBCNSU2(IBCNS,1,1,.IBSEL,0,1) Q:IBQUIT  ..I '$O(^TMP($J,"IBSEL",0)) K ^TMP("IBINC",$J,IBIC,IBCNS) S IBAI=0 Q  ..;  ..; - set plans into an array  ..S IBPN=0 F  S IBPN=$O(^TMP($J,"IBSEL",IBPN)) Q:'IBPN  S ^TMP("IBINC",$J,IBIC,IBCNS,IBPN)=""  ; STARTQ K IBCNS,IBIC,IBJJ,IBCT,IBLCT,IBOK,IBPN,IBSEL,VAUTI,VAUTP,^TMP($J,"IBSEL")  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOPP1** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.11.1, 2.6.11.2, 2.6.11.3, 2.6.11.17 | | | | | | | | |
| **Related Options** | List Plans by Insurance Company [IBCN LIST PLANS BY INS CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCOPP | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOPP1 ;ALB/NLR - LIST INS. PLANS BY CO. (DRIVER 1) ; 15-SEP-94   ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28\*\*; 21-MAR-94  ; SELR() ; Prompt user to select report type  ;  ; IBV1=0 -- list insurance plans by company  ; IBV1=1 -- list insurance plans by company with subscriber information  ;  N IBV1  S DIR(0)="SA^1:1. List Insurance Plans by Company;2:2. List Insurance Plans by Company With Subscriber Information"  S DIR("A")=" SELECT REPORT (1 OR 2): "  S DIR("A",1)="1. List Insurance Plans by Company"  S DIR("A",2)="2. List Insurance Plans by Company With Subscriber Information"  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV1=-1 G SELRQ  S IBV1=(+Y=2) K Y SELRQ Q IBV1  ; SELI() ; Prompt user to select all or subset of insurance companies   ; Count ins. companies with plans  ;  ; IBV2=0 -- user selects insurance companies  ; IBV2=1 -- run report for all insurance companies with plans  ;  N A,B,IBV2  S (A,B)=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S B=B+1  S DIR(0)="SA^1:1. List All "\_B\_" Ins. Companies;2:2. List Only Ins. Companies That You Select"  W !!," There are "\_B\_" insurance companies associated with plans.",!  S DIR("A",1)="1. List All "\_B\_" Ins. Companies"  S DIR("A",2)="2. List Only Ins. Companies That You Select"  S DIR("A")=" SELECT 1 or 2: "  S DIR("?",1)="Enter a code from the list: 1 or 2. Only insurance"  S DIR("?")="companies with one or more plans can be selected."  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV2=-1 G SELIQ  S IBV2=(+Y=1) K Y SELIQ Q IBV2  ; SELP() ; Prompt user to select all or subset of plans  ;  ; IBV3=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies  ; IBV3=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ;  N IBV3  S DIR(0)="YO",DIR("A")="There are "\_$P(^IBA(355.3,0),"^",4)\_" plans. List all plans for each company",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will list all of the plans for each company."  S DIR("?",2)="If you selected 2. List Insurance Plans by Company With Subscriber"  S DIR("?",3)="Information and 1. List All "\_$P(^IBA(355.3,0),"^",4)\_" Ins. Companies,"  S DIR("?",4)="this will result in the most complete report possible. However, it"  S DIR("?",5)="may take awhile to run. If you say no, you must make plan selections"  S DIR("?")="for each individual company (anywhere from one plan to all)."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV3=-1 G SELPQ  S IBV3=+Y K Y SELPQ Q IBV3 |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOPP1 ;ALB/NLR - LIST INS. PLANS BY CO. (DRIVER 1) ; 15-SEP-94   ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28\*\*; 21-MAR-94  ; SELR() ; Prompt user to select report type  ;  ; IBV1=0 -- list insurance plans by company  ; IBV1=1 -- list insurance plans by company with subscriber information  ;  N IBV1  S DIR(0)="SA^1:1. List Insurance Plans by Company;2:2. List Insurance Plans by Company With Subscriber Information"  S DIR("A")=" SELECT REPORT (1 OR 2): "  S DIR("A",1)="1. List Insurance Plans by Company"  S DIR("A",2)="2. List Insurance Plans by Company With Subscriber Information"  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV1=-1 G SELRQ  S IBV1=(+Y=2) K Y SELRQ Q IBV1  ; SELI() ; Prompt user to select all or subset of insurance companies   ; Count ins. companies with plans  ;  ; IBV2=0 -- user selects insurance companies  ; IBV2=1 -- run report for all insurance companies with plans  ;  N A,B,IBV2  S (A,B)=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S B=B+1  S DIR(0)="SA^1:1. List All "\_B\_" Ins. Companies;2:2. List Only Ins. Companies That You Select"  W !!," There are "\_B\_" insurance companies associated with plans.",!  S DIR("A",1)="1. List All "\_B\_" Ins. Companies"  S DIR("A",2)="2. List Only Ins. Companies That You Select"  S DIR("A")=" SELECT 1 or 2: "  S DIR("?",1)="Enter a code from the list: 1 or 2. Only insurance"  S DIR("?")="companies with one or more plans can be selected."  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV2=-1 G SELIQ  S IBV2=(+Y=1) K Y SELIQ Q IBV2 ;  **SELA() ; Prompt user to select report type ; 2.6.11.1, 2.6.11.17 add selection of Active/Inactive/Both for insurance companies  ;  ; IBV4=0 -- Select ACTIVE Insurance Companies  ; IBV4=1 -- Select INACTIVE Insurance Companies  ; IBV4=2 -- Select BOTH  ;  N IBV4  S DIR(0)="SA^1:1. Select ACTIVE Insurance Companies?;2:2. Select INACTIVE Insurance Companies?;3:3. Select BOTH?"  S DIR("A")=" SELECT REPORT (1 or 2 or 3): selecting 1 or 2 would filter the resulting insurance company list to display the appropriate results to select from. What user selects here will apply for all companies during rinse & repeat cycle.>"  S DIR("A",1)="1. Select ACTIVE Insurance Companies?"  S DIR("A",2)="2. Select INACTIVE Insurance Companies?"  S DIR("A",3)="3. Select BOTH?"  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV4=-1 G SELAQ  S IBV4=+Y K Y SELAQ Q IBV4  ;**   ; SELP() ; Prompt user to select all or subset of plans  ;  ; IBV3=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies  ; IBV3=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ;  N IBV3  S DIR(0)="YO",DIR("A")="There are "\_$P(^IBA(355.3,0),"^",4)\_" plans. List all plans for each company",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will list all of the plans for each company."  S DIR("?",2)="If you selected 2. List Insurance Plans by Company With Subscriber"  S DIR("?",3)="Information and 1. List All "\_$P(^IBA(355.3,0),"^",4)\_" Ins. Companies,"  S DIR("?",4)="this will result in the most complete report possible. However, it"  S DIR("?",5)="may take awhile to run. If you say no, you must make plan selections"  S DIR("?")="for each individual company (anywhere from one plan to all)."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV3=-1 G SELPQ  S IBV3=+Y K Y SELPQ Q IBV3  ;  **SELPA() ; Prompt user to select report type ; 2.6.11.3, 2.6.11.17 add selection of Active/Inactive/Both for insurance companies  ;  ; IBV5=0 -- Select ACTIVE Group Plans  ; IBV5=1 -- Select INACTIVE Group Plans  ; IBV5=2 -- Select BOTH  ;  N IBV4  S DIR(0)="SA^1:1. Select ACTIVE Insurance Companies?;2:2. Select INACTIVE Insurance Companies?;3:3. Select BOTH?"  S DIR("A")=" SELECT REPORT (1 or 2 or 3): selecting 1 or 2 would filter the resulting insurance company list to display the appropriate results to select from. What user selects here will apply for all Group Plan during rinse & repeat cycle.>"  S DIR("A",1)="1. Select ACTIVE Group Plans?"  S DIR("A",2)="2. Select INACTIVE Group Plans?"  S DIR("A",3)="3. Select BOTH?"  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV5=-1 G SELPAQ  S IBV4=+Y K Y SELPAQ Q IBV4  ;** |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOPP2** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.11.4, 2.6.11.5, 2.6.11.7, 2.6.11.8, 2.6.11.9, 2.6.11.10, 2.6.11.11, 2.6.11.12, 2.6.11.13, 2.6.11.14 | | | | | | | | |
| **Related Options** | List Plans by Insurance Company [IBCN LIST PLANS BY INS CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT  IBCOPP | | | | | IBCOPP3  PT^IBEFUNC  DAT1^IBOUTL  DAT3^IBOUTL | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOPP2 ;ALB/NLR - LIST INS. PLANS BY CO. (COMPILE) ; 06-SEP-94 V ;;2.0;INTEGRATED BILLING;\*\*28,62,93\*\*;21-MAR-94  ; EN ; Queued Entry Point for Report.  ; Required variable input: IBAI, IBAPL, IBAPA  ; ^TMP("IBINC",$J) required if all companies and plans not selected  ;  ; - compile report data  S IBI=0 K ^TMP($J,"PR"),^TMP($J,"PL")  ;  ; - user wanted all companies and plans  I IBAI,IBAPL D  G PRINT  .S IBIC1="" F  S IBIC1=$O(^DIC(36,"B",IBIC1)) Q:IBIC1=""  D  ..S IBCNS=0 F  S IBCNS=$O(^DIC(36,"B",IBIC1,IBCNS)) Q:'IBCNS  I $D(^IBA(355.3,"B",IBCNS)) S IBIC=IBIC1 D GATH  ;  ; - user selected companies or plans  S IBIC="" F  S IBIC=$O(^TMP("IBINC",$J,IBIC)) Q:IBIC=""  D  .S IBCNS=0 F  S IBCNS=$O(^TMP("IBINC",$J,IBIC,IBCNS)) Q:'IBCNS  D GATH  ; PRINT ; - print report  D ^IBCOPP3  K ^TMP($J,"PR"),^TMP("IBINC",$J)  ;  I $D(ZTQUEUED) S ZTREQ="@" Q  D ^%ZISC  K IBI,IBIC,IBIC1,IBCNS,IBCPT,IBCPS,IBCST,IBCSS  Q  ;  ; GATH ; Gather all data for a company.  S IBI=IBI+1,(IBCPT,IBCPS,IBCST,IBCSS)=0 ; initialize counters  D COMP ; gather company info  D PLAN ; gather plan info  ;  ; - set final company info  S ^TMP($J,"PR",IBI)=$$COMPINF(IBCNS)\_"^"\_IBCPT\_"^"\_IBCST\_"^"\_IBCPS\_"^"\_IBCSS  K ^TMP($J,"PL")  Q  ;  ; COMP ; Gather Company counts and subscription information, if necessary  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; initialized counters, plus the 'Plan' array (^TMP("IBINC",$J))  ;   S DFN=0 F  S DFN=$O(^DPT("AB",IBCNS,DFN)) Q:'DFN  D  .S IBCDFN=0 F  S IBCDFN=$O(^DPT("AB",IBCNS,DFN,IBCDFN)) Q:'IBCDFN  D  ..;  ..; - set company subscriber count; plan subscriber counts if necessary  ..S IBIND=$G(^DPT(DFN,.312,+IBCDFN,0)) Q:+IBIND'=IBCNS  ..S IBPTR=+$P(IBIND,"^",18)  ..S IBCST=IBCST+1  ..I 'IBAPL,'$D(^TMP("IBINC",$J,IBIC,IBCNS,IBPTR)) Q  ; not a selected plan  ..S IBCSS=IBCSS+1,^(IBPTR)=$G(^TMP($J,"PL",IBPTR))+1  ..Q:'IBAPA  ; policy information not selected  ..;  ..; - gather demographic/policy information  ..S X=$$PT^IBEFUNC(DFN)  ..S IBNAM=$E($S($P(X,"^")]"":$P(X,"^"),1:"<Pt. "\_DFN\_" Name Missing>")\_$J("",25),1,25)\_" ("\_$E(X)\_$P(X,"^",3)\_")"  ..S IBDOB=$$DAT3^IBOUTL($P($G(^DPT(DFN,0)),"^",3))  ..S IBWI=$P(IBIND,"^",6),IBWI=$S(IBWI="v":"VET",IBWI="s":"SPO",IBWI="o":"OTH",1:"<UNK>")  ..S VAOA("A")=$S(IBWI="SPO":6,1:5) D OAD^VADPT  ..;  ..; - build detail line  ..S IBX=IBNAM\_U\_IBDOB\_U\_$E(VAOA(9),1,18)\_U\_$S($P(IBIND,"^",2)]"":$E($P(IBIND,"^",2),1,17),1:"<NO SUBS ID>")  ..S IBX=IBX\_U\_IBWI\_U\_$$DAT1^IBOUTL($P(IBIND,"^",8))\_U\_$$DAT1^IBOUTL($P(IBIND,"^",4))  ..S X=0,Y="" F  S Y=$O(^IBA(355.5,"APPY",DFN,IBPTR,Y)) Q:Y=""  I $O(^(Y,0))=IBCDFN S X=1 Q  ..S ^TMP($J,"PR",IBI,IBPTR,IBNAM\_"@@"\_DFN\_"@@"\_IBCDFN)=IBX\_"^"\_X  ;  K DFN,IBCDFN,IBIND,IBPTR,IBNAM,IBDOB,IBWI,IBX,X,VAOA,VA,VAERR,Y  Q  ; PLAN ; Gather Insurance Plan information, if necessary  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; initialized counters, plus the 'Plan' array (^TMP("IBINC",$J))  ;   S IBPTR=0 F  S IBPTR=$O(^IBA(355.3,"B",IBCNS,IBPTR)) Q:'IBPTR  D  .S IBCPT=IBCPT+1  .I 'IBAPL,'$D(^TMP("IBINC",$J,IBIC,IBCNS,IBPTR)) Q  ; not a selected plan  .S IBCPS=IBCPS+1  .S ^TMP($J,"PR",IBI,IBPTR)=$$PLANINF(IBPTR)\_"^"\_+$G(^TMP($J,"PL",IBPTR))  K IBPTR  Q  ; PLANINF(PLAN) ; Return formatted Insurance Plan information.  ; Input: PLAN -- Pointer to the plan in file #355.3  ; Output: plan number ^ name ^ grp/ind ^ act/inact  ;  N ACT,NAME,NUM,TY,X  S X=$G(^IBA(355.3,PLAN,0))  S TY=$S($P(X,"^",2):"GRP",1:"IND")  S NAME=$P(X,"^",3) S:NAME="" NAME="<NO GROUP NAME>"  S NUM=$P(X,"^",4) S:NUM="" NUM="<NO GROUP NUMBER>"  S ACT=$S($P(X,"^",11):"IN",1:"")\_"ACTIVE"  Q NUM\_"^"\_NAME\_"^"\_TY\_"^"\_ACT\_"^"\_$S($D(^IBA(355.4,"APY",PLAN))>0:"YES",1:"NO")\_"^"\_$S($D(^IBA(355.5,"B",PLAN))>0:"YES",1:"NO")  ; COMPINF(IBCNS) ; Return formatted Insurance Company information  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; Output: company name ^ addr ^ city/st/zip ^ phone ^ precert ^ act?  ;  N ST,X,X0,X11,X13,Z  S X0=$G(^DIC(36,IBCNS,0)),X11=$G(^(.11)),X13=$G(^(.13)),Z=$P(X11,"^",6)  S ST=$S($P(X11,"^",5):$P($G(^DIC(5,$P(X11,"^",5),0)),"^",2),1:"<STATE MISSING>")  S X="Ins. Co.: "\_$E($P(X0,"^"),1,25)  S X=X\_U\_$S($P(X11,"^")'="":$P(X11,"^"),1:"<Street Addr. 1 Missing>")  S X=X\_U\_$P(X11,"^",4)\_", "\_ST\_" "\_$E(Z,1,5)\_$S($E(Z,6,9)]"":"-"\_$E(Z,6,9),1:"")  S X=X\_U\_"Phone: "\_$P(X13,"^")\_U\_"Precert Phone: "\_$P(X13,"^",3)  Q X\_U\_$S($P(X0,"^",5):"IN",1:"")\_"ACTIVE COMPANY" |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOPP2 ;ALB/NLR - LIST INS. PLANS BY CO. (COMPILE) ; 06-SEP-94 V ;;2.0;INTEGRATED BILLING;\*\*28,62,93\*\*;21-MAR-94  ; EN ; Queued Entry Point for Report.  ; Required variable input: IBAI, IBAPL, IBAPA  ; ^TMP("IBINC",$J) required if all companies and plans not selected  ;  ; - compile report data  S IBI=0 K ^TMP($J,"PR"),^TMP($J,"PL")  ;  ; - user wanted all companies and plans  I IBAI,IBAPL D  G PRINT  .S IBIC1="" F  S IBIC1=$O(^DIC(36,"B",IBIC1)) Q:IBIC1=""  D  ..S IBCNS=0 F  S IBCNS=$O(^DIC(36,"B",IBIC1,IBCNS)) Q:'IBCNS  I $D(^IBA(355.3,"B",IBCNS)) S IBIC=IBIC1 D GATH  ;  ; - user selected companies or plans  S IBIC="" F  S IBIC=$O(^TMP("IBINC",$J,IBIC)) Q:IBIC=""  D  .S IBCNS=0 F  S IBCNS=$O(^TMP("IBINC",$J,IBIC,IBCNS)) Q:'IBCNS  D GATH  ; PRINT ; - print report  D ^IBCOPP3  K ^TMP($J,"PR"),^TMP("IBINC",$J)  ;  I $D(ZTQUEUED) S ZTREQ="@" Q  D ^%ZISC  K IBI,IBIC,IBIC1,IBCNS,IBCPT,IBCPS,IBCST,IBCSS  Q  ;  ; GATH ; Gather all data for a company.  S IBI=IBI+1,(IBCPT,IBCPS,IBCST,IBCSS)=0 ; initialize counters  D COMP ; gather company info  D PLAN ; gather plan info  ;  ; - set final company info  S ^TMP($J,"PR",IBI)=$$COMPINF(IBCNS)\_"^"\_IBCPT\_"^"\_IBCST\_"^"\_IBCPS\_"^"\_IBCSS  K ^TMP($J,"PL")  Q  ;  ; COMP ; Gather Company counts and subscription information, if necessary  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; initialized counters, plus the 'Plan' array (^TMP("IBINC",$J))  ;   S DFN=0 F  S DFN=$O(^DPT("AB",IBCNS,DFN)) Q:'DFN  D  .S IBCDFN=0 F  S IBCDFN=$O(^DPT("AB",IBCNS,DFN,IBCDFN)) Q:'IBCDFN  D  ..;  ..; - set company subscriber count; plan subscriber counts if necessary **..;2.6.11.4 - add capture of Timely Filing (insurance company level)**  ..S IBIND=$G(^DPT(DFN,.312,+IBCDFN,0)) Q:+IBIND'=IBCNS  ..S IBPTR=+$P(IBIND,"^",18)  ..S IBCST=IBCST+1  ..I 'IBAPL,'$D(^TMP("IBINC",$J,IBIC,IBCNS,IBPTR)) Q  ; not a selected plan  ..S IBCSS=IBCSS+1,^(IBPTR)=$G(^TMP($J,"PL",IBPTR))+1  ..Q:'IBAPA  ; policy information not selected  ..;  ..; - gather demographic/policy information **..;2.6.11.4 - add capture of Timely Filing, Electronic Plan Type, Type of Plan information**  ..S X=$$PT^IBEFUNC(DFN)  **..;2.6.11.7 - Remove Parenthesis around Subscriber ID and add as separate ‘^’ piece**   ..S IBNAM=$E($S($P(X,"^")]"":$P(X,"^"),1:"<Pt. "\_DFN\_" Name Missing>")\_$J("",25),1,25)\_" ("\_$E(X)\_$P(X,"^",3)\_")"  ..S IBDOB=$$DAT3^IBOUTL($P($G(^DPT(DFN,0)),"^",3)) **..;2.6.11.14 - Remove Employer**   ..S IBWI=$P(IBIND,"^",6),IBWI=$S(IBWI="v":"VET",IBWI="s":"SPO",IBWI="o":"OTH",1:"<UNK>")  ..S VAOA("A")=$S(IBWI="SPO":6,1:5) D OAD^VADPT  ..;  ..; - build detail line  **..;2.6.11.5 - add Patient ID  ..;2.6.11.8 - Add last 4 numbers of SSN as separate ‘^’ piece**  ..S IBX=IBNAM\_U\_IBDOB\_U\_$E(VAOA(9),1,18)\_U\_$S($P(IBIND,"^",2)]"":$E($P(IBIND,"^",2),1,17),1:"<NO SUBS ID>")  ..S IBX=IBX\_U\_IBWI\_U\_$$DAT1^IBOUTL($P(IBIND,"^",8))\_U\_$$DAT1^IBOUTL($P(IBIND,"^",4)) **..;2.6.11.15 - Remove next two lines**  **~~..S X=0,Y="" F  S Y=$O(^IBA(355.5,"APPY",DFN,IBPTR,Y)) Q:Y=""  I $O(^(Y,0))=IBCDFN S X=1 Q  ..S ^TMP($J,"PR",IBI,IBPTR,IBNAM\_"@@"\_DFN\_"@@"\_IBCDFN)=IBX\_"^"\_X~~**  ;  K DFN,IBCDFN,IBIND,IBPTR,IBNAM,IBDOB,IBWI,IBX,X,VAOA,VA,VAERR,Y  Q  ; PLAN ; Gather Insurance Plan information, if necessary  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; initialized counters, plus the 'Plan' array (^TMP("IBINC",$J))  ;   S IBPTR=0 F  S IBPTR=$O(^IBA(355.3,"B",IBCNS,IBPTR)) Q:'IBPTR  D  .S IBCPT=IBCPT+1  .I 'IBAPL,'$D(^TMP("IBINC",$J,IBIC,IBCNS,IBPTR)) Q  ; not a selected plan  .S IBCPS=IBCPS+1  .S ^TMP($J,"PR",IBI,IBPTR)=$$PLANINF(IBPTR)\_"^"\_+$G(^TMP($J,"PL",IBPTR))  K IBPTR  Q  ; PLANINF(PLAN) ; Return formatted Insurance Plan information.  ; Input: PLAN -- Pointer to the plan in file #355.3  ; Output: plan number ^ name ^ grp/ind ^ act/inact  **;2.6.11.4 - add capture of Timely Filing, Electronic Plan Type, Type of Plan information**  ;  N ACT,NAME,NUM,TY,X  S X=$G(^IBA(355.3,PLAN,0))  S TY=$S($P(X,"^",2):"GRP",1:"IND")  S NAME=$P(X,"^",3) S:NAME="" NAME="<NO GROUP NAME>"  S NUM=$P(X,"^",4) S:NUM="" NUM="<NO GROUP NUMBER>"  S ACT=$S($P(X,"^",11):"IN",1:"")\_"ACTIVE"  Q NUM\_"^"\_NAME\_"^"\_TY\_"^"\_ACT\_"^"\_$S($D(^IBA(355.4,"APY",PLAN))>0:"YES",1:"NO")\_"^"\_$S($D(^IBA(355.5,"B",PLAN))>0:"YES",1:"NO")  ; COMPINF(IBCNS) ; Return formatted Insurance Company information  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; Output: company name ^ addr ^ city/st/zip ^ phone ^ precert ^ act? **;2.6.11.9 Add Active/Inactive flag  ;2.6.11.10 Remove Phone  ;2.6.11.11 Remove Precert Phone  ;2.6.11.12 Remove Active/Inactive Indicator  ;2.6.11.13 Remove Group or Ind**   ;  N ST,X,X0,X11,X13,Z  S X0=$G(^DIC(36,IBCNS,0)),X11=$G(^(.11)),X13=$G(^(.13)),Z=$P(X11,"^",6)  S ST=$S($P(X11,"^",5):$P($G(^DIC(5,$P(X11,"^",5),0)),"^",2),1:"<STATE MISSING>")  S X="Ins. Co.: "\_$E($P(X0,"^"),1,25)  S X=X\_U\_$S($P(X11,"^")'="":$P(X11,"^"),1:"<Street Addr. 1 Missing>")  S X=X\_U\_$P(X11,"^",4)\_", "\_ST\_" "\_$E(Z,1,5)\_$S($E(Z,6,9)]"":"-"\_$E(Z,6,9),1:"")  S X=X\_U\_"Phone: "\_$P(X13,"^")\_U\_"Precert Phone: "\_$P(X13,"^",3)  Q X\_U\_$S($P(X0,"^",5):"IN",1:"")\_"ACTIVE COMPANY" |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOPP3** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.11.4, 2.6.11.5, 2.6.11.7, 2.6.11.8, 2.6.11.12, 2.6.11.13, 2.6.11.14, 2.6.11.15, 2.6.11.16, 2.6.11.18 | | | | | | | | |
| **Related Options** | List Plans by Insurance Company [IBCN LIST PLANS BY INS CO] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT2  IBCOPP2 | | | | | DAT2^IBOUTL | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOPP3 ;ALB/NLR - LIST INS. PLANS BY CO. (PRINT) ; 04-OCT-94  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28\*\*; 21-MAR-94  ;  ; Print the report.  ; Required Input: Global print array ^TMP($J,"PR"  ; local variable IBAPA  ;   S (IBI,IBQUIT,IBPAG)=0  D NOW^%DTC S IBHDT=$$DAT2^IBOUTL($E(%,1,12))  F  S IBI=$O(^TMP($J,"PR",IBI)) Q:'IBI  S IBC=$G(^(IBI)) D COMP D  Q:IBQUIT  .S IBP=0 F  S IBP=$O(^TMP($J,"PR",IBI,IBP)) Q:'IBP  S IBPD=$G(^(IBP)) D  Q:IBQUIT  ..I $Y>(IOSL-$S(IBAPA:9,1:5)) D PAUSE Q:IBQUIT  D COMP  ..D PLAN  ..I IBAPA S IBS="" F  S IBS=$O(^TMP($J,"PR",IBI,IBP,IBS)) Q:IBS=""  S IBSD=$G(^(IBS)) D SUBS Q:IBQUIT  .Q:IBQUIT  .;  .; - print company totals  .I $Y>(IOSL-4) D PAUSE Q:IBQUIT  D COMP,PLAN  .W !!?90,"Number of Plans Selected = ",$P(IBC,"^",9),!?76,"Total Subscribers Under Selected Plans = ",$P(IBC,"^",10)  .D PAUSE  ;  K IBJJ,IBI,IBQUIT,IBPAG,IBHDT,IBC,IBP,IBPD,IBS,IBSD  Q  ;  ; COMP ; Print Company header  I $E(IOST,1,2)="C-"!(IBPAG) W @IOF  S IBPAG=IBPAG+1  W !,"LIST OF PLANS BY INSURANCE COMPANY"  W:IBAPA " WITH SUBSCRIBER INFORMATION"  W ?IOM-34,IBHDT,?IOM-10,"Page: ",IBPAG  W !,$TR($J(" ",IOM)," ","-")  ;  ; - sub-header  W !?1,$P(IBC,"^"),?45,$P(IBC,"^",4),?105,$P(IBC,"^",6)  W !?11,$P(IBC,"^",2),?45,$P(IBC,"^",5),?105,"PLAN TOTAL= ",$P(IBC,"^",7)  W !?11,$P(IBC,"^",3),?99,"SUBSCRIBER TOTAL= ",$P(IBC,"^",8)  W:IBAPA !!?95,"WHOSE",?127,"BEN.",!?3,"SUBSCRIBER NAME/ID",?40,"DOB",?54,"EMPLOYER",?76,"SUBSCR ID",?95,"INS",?105,"EFF DATE",?117,"EXP DATE",?127,"USED?"  W:'IBAPA !!?5,"GROUP NUMBER",?32,"GROUP NAME",?62,"GROUP OR IND",?77,"ACTIVE/INACTIVE",?96,"SUBSCRIBERS",?110,"ANN. BEN? BEN. USED?"  Q  ; PLAN ; Print plan information.  I IBAPA D   .W !!?3,"GROUP #: ",$P(IBPD,U),?40,"ANNUAL BENEFITS ON FILE: ",$P(IBPD,U,5)  .W !?5,"GROUP NAME: ",$P(IBPD,U,2),?42,"BENEFITS USED ON FILE: ",$P(IBPD,U,6)  .W !?7,"GROUP OR IND: ",$P(IBPD,U,3),!?9,"ACTIVE?: ",$P(IBPD,U,4),!?11,"NO. SUBSCRIBERS: ",$P(IBPD,U,7)  I 'IBAPA W !!?5,$P(IBPD,U),?32,$P(IBPD,U,2),?62,$P(IBPD,U,3),?77,$P(IBPD,U,4),?100,$P(IBPD,U,7),?113,$P(IBPD,U,5),?124,$P(IBPD,U,6)  Q  ; SUBS ; Print subscriber information.  I $Y>(IOSL-4) D PAUSE Q:IBQUIT  D COMP,PLAN  W !?3,$P(IBSD,"^"),?40,$P(IBSD,"^",2),?54,$P(IBSD,"^",3),?76,$P(IBSD,"^",4),?95,$P(IBSD,"^",5)  W ?105,$P(IBSD,"^",6),?117,$P(IBSD,"^",7),?128,$S($P(IBSD,"^",8):"YES",1:"NO")  Q  ; PAUSE ; Pause for screen output.  Q:$E(IOST,1,2)'["C-"  F IBJJ=$Y:1:(IOSL-7) W !  S DIR(0)="E" D ^DIR K DIR I $D(DIRUT)!($D(DUOUT)) S IBQUIT=1 K DIRUT,DTOUT,DUOUT  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOPP3 ;ALB/NLR - LIST INS. PLANS BY CO. (PRINT) ; 04-OCT-94  ;;Version 2.0 ; INTEGRATED BILLING ;\*\*28\*\*; 21-MAR-94  ;  ; Print the report.  ; Required Input: Global print array ^TMP($J,"PR"  ; local variable IBAPA  ;   S (IBI,IBQUIT,IBPAG)=0  D NOW^%DTC S IBHDT=$$DAT2^IBOUTL($E(%,1,12))  F  S IBI=$O(^TMP($J,"PR",IBI)) Q:'IBI  S IBC=$G(^(IBI)) D COMP D  Q:IBQUIT  .S IBP=0 F  S IBP=$O(^TMP($J,"PR",IBI,IBP)) Q:'IBP  S IBPD=$G(^(IBP)) D  Q:IBQUIT  ..I $Y>(IOSL-$S(IBAPA:9,1:5)) D PAUSE Q:IBQUIT  D COMP  ..D PLAN  ..I IBAPA S IBS="" F  S IBS=$O(^TMP($J,"PR",IBI,IBP,IBS)) Q:IBS=""  S IBSD=$G(^(IBS)) D SUBS Q:IBQUIT  .Q:IBQUIT  .;  .; - print company totals  .I $Y>(IOSL-4) D PAUSE Q:IBQUIT  D COMP,PLAN  .W !!?90,"Number of Plans Selected = ",$P(IBC,"^",9),!?76,"Total Subscribers Under Selected Plans = ",$P(IBC,"^",10)  .D PAUSE **; 2.6.11.18 Add End of report indicator**   ;  K IBJJ,IBI,IBQUIT,IBPAG,IBHDT,IBC,IBP,IBPD,IBS,IBSD  Q  ;  ; COMP ; Print Company header  I $E(IOST,1,2)="C-"!(IBPAG) W @IOF  S IBPAG=IBPAG+1  W !,"LIST OF PLANS BY INSURANCE COMPANY"  W:IBAPA " WITH SUBSCRIBER INFORMATION"  W ?IOM-34,IBHDT,?IOM-10,"Page: ",IBPAG  W !,$TR($J(" ",IOM)," ","-")  ;  ; - sub-header  W !?1,$P(IBC,"^"),?45,$P(IBC,"^",4),?105,$P(IBC,"^",6)  W !?11,$P(IBC,"^",2),?45,$P(IBC,"^",5),?105,"PLAN TOTAL= ",$P(IBC,"^",7)  W !?11,$P(IBC,"^",3),?99,"SUBSCRIBER TOTAL= ",$P(IBC,"^",8) **; 2.6.11.4 Add Electronic Plan Type, Timely Filing Group Plan Type column headers**  **; 2.6.11.5 Add Patient ID column header**  **; 2.6.11.7 Change Subscriber Name/ID to two columns**  **; 2.6.11.8 Add SSN col header**  **; 2.6.11.14 Remove EMPLOYER col header**   W:IBAPA !!?95,"WHOSE",?127,"BEN.",!?3,"SUBSCRIBER NAME/ID",?40,"DOB",?54,"EMPLOYER",?76,"SUBSCR ID",?95,"INS",?105,"EFF DATE",?117,"EXP DATE",?127,"USED?" **; 2.6.11.12 Remove ACTIVE/INACTIVE col header**  **; 2.6.11.13 Remove GROUP OR IND col header**  **; 2.6.11.15 Remove BEN USED? col header**  **; 2.6.11.16 Remove ANN BEN? col header**   W:'IBAPA !!?5,"GROUP NUMBER",?32,"GROUP NAME",?62,"GROUP OR IND",?77,"ACTIVE/INACTIVE",?96,"SUBSCRIBERS",?110,"ANN. BEN? BEN. USED?"  Q  ; PLAN ; Print plan information.  I IBAPA D **; 2.6.11.5 Add Patient ID**  .W !!?3,"GROUP #: ",$P(IBPD,U),?40,"ANNUAL BENEFITS ON FILE: ",$P(IBPD,U,5)  .W !?5,"GROUP NAME: ",$P(IBPD,U,2),?42,"BENEFITS USED ON FILE: ",$P(IBPD,U,6)  .W !?7,"GROUP OR IND: ",$P(IBPD,U,3),!?9,"ACTIVE?: ",$P(IBPD,U,4),!?11,"NO. SUBSCRIBERS: ",$P(IBPD,U,7)  I 'IBAPA W !!?5,$P(IBPD,U),?32,$P(IBPD,U,2),?62,$P(IBPD,U,3),?77,$P(IBPD,U,4),?100,$P(IBPD,U,7),?113,$P(IBPD,U,5),?124,$P(IBPD,U,6)  Q  ; SUBS ; Print subscriber information.  I $Y>(IOSL-4) D PAUSE Q:IBQUIT  D COMP,PLAN  W !?3,$P(IBSD,"^"),?40,$P(IBSD,"^",2),?54,$P(IBSD,"^",3),?76,$P(IBSD,"^",4),?95,$P(IBSD,"^",5)  W ?105,$P(IBSD,"^",6),?117,$P(IBSD,"^",7),?128,$S($P(IBSD,"^",8):"YES",1:"NO")  Q  ; PAUSE ; Pause for screen output.  Q:$E(IOST,1,2)'["C-"  F IBJJ=$Y:1:(IOSL-7) W !  S DIR(0)="E" D ^DIR K DIR I $D(DIRUT)!($D(DUOUT)) S IBQUIT=1 K DIRUT,DTOUT,DUOUT  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCEMMR** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.13.1, 2.6.13.2, 2.6.13.3, 2.6.13.4, 2.6.13.5 , 2.6.13.6, 2.6.13.7, 2.6.13.8, 2.6.13.9 | | | | | | | | |
| **Related Options** | Patients Without MEDICARE (WNR) Insurance [IBCEM PATIENTS] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | ALLWNR^IBCNS1 MCRWNR^IBEFUNC  LAST^IBSDU  NEXT^IBSDU | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCEMMR ;ALB/ESG - IB MRA Report of Patients w/o Medicare WNR ;20-NOV-2003  ;;2.0;INTEGRATED BILLING;\*\*155,366\*\*;21-MAR-94;Build 3  ;  ; Find patients with Medicare supplemental insurance or Medigap  ; insurance (etc.) but who do not have MEDICARE (WNR) on file as  ; one of their insurances.  ;  Q  ; EN ; Entry Point  NEW IBMSORT  D SORT I 'IBMSORT G EX  D DEVICE EX ; Exit Point  Q  ; SORT ; Ask user how to sort the report  NEW CH,DIR,X,Y,DIRUT,DIROUT  W @IOF,!?20,"Patients Without MEDICARE (WNR) Insurance"  W !!?2,"This option finds patients who do not have active MEDICARE (WNR) insurance,"  W !?2,"but who do have active insurance with a Plan Type of Medigap, Carve-Out, or"  W !?2,"Medicare Secondary. In these cases, MEDICARE (WNR) should be primary."  W !!?2,"The insurances for all living patients will be analyzed, but"  W !?2,"you can determine how this information will be sorted."  S IBMSORT=""  W !  S CH="1:Patient Name;2:SSN - Last 4 Digits;3:Insurance Company;"  S CH=CH\_"4:Type of Plan;5:Appointment Date"  S DIR(0)="SO^"\_CH  S DIR("A")="Please enter the Sort Criteria"  S DIR("B")="Patient Name"  D ^DIR K DIR  I 'Y G SORTX  S IBMSORT=Y SORTX ;  Q  ; COMPILE ; Entry point for both background and foreground task execution  ;  NEW RTN,DFN,CNT,MS,DPT,PTNM,SSN,APPT,APDTE,A  NEW INS,GRP,PLN,INSNM,PLNTYP,SORT,X,IBNEXT  S RTN="IBCEMMR"  K ^TMP($J,RTN),^("IBCEPT"),^("IBSDNEXT"),^("IBDPT"),^("IBLAST")  S DFN=" ",CNT=0  F  S DFN=$O(^DPT(DFN),-1) Q:'DFN!($G(ZTSTOP)) D  . S CNT=CNT+1  . I '$D(ZTQUEUED),CNT#500=0 U IO(0) W "." U IO  . I $D(ZTQUEUED),CNT#500=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . I $P($G(^DPT(DFN,.35)),U,1) Q           ; date of death  . I '$$PTINS(DFN,.MS) Q                   ; eligible for report  . S ^TMP($J,"IBNEXT",DFN)=""  . S ^TMP($J,"IBLAST",DFN)=""  . S ^TMP($J,"IBDPT",DFN)=""  ;  S X=$$NEXT^IBSDU("^TMP($J,""IBNEXT"",")  S X=$$LAST^IBSDU("^TMP($J,""IBLAST"",")  ;  S DFN=0 F  S DFN=$O(^TMP($J,"IBDPT",DFN)) Q:'DFN!($G(ZTSTOP)) D  . I '$D(ZTQUEUED),CNT#500=0 U IO(0) W "." U IO  . I $D(ZTQUEUED),CNT#500=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . I '$$PTINS(DFN,.MS) ; get MS data  . S DPT=$G(^DPT(DFN,0))  . S PTNM=$P(DPT,U,1)  . I PTNM="" S PTNM="~UNKNOWN"  . S SSN=$E($P(DPT,U,9),6,99)\_" "  . I SSN="" S SSN="~UNK"  . S (APPT,IBNEXT)=$G(^TMP($J,"IBNEXT",DFN),"UNKNOWN")  . I 'APPT S APPT=$G(^TMP($J,"IBLAST",DFN),"UNKNOWN")  . S APDTE=$S(APPT:$$FMTE^XLFDT($P(APPT,"."),"2Z"),$L(IBNEXT):IBNEXT,$L(APPT):APPT,1:"N/A")  . S APPT=+APPT  . S A=0 F  S A=$O(MS(A)) Q:'A  D  .. S INS=+$P(MS(A),U,1),GRP=+$P(MS(A),U,2)  .. S PLN=+$P(MS(A),U,3)  .. S INSNM=$P($G(^DIC(36,INS,0)),U,1)  .. I INSNM="" S INSNM="~UNKNOWN"  .. S PLNTYP=$P($G(^IBE(355.1,PLN,0)),U,1)  .. I PLNTYP="" S PLNTYP="~UNKNOWN"  .. S SORT=$S(IBMSORT=1:PTNM,IBMSORT=2:SSN,IBMSORT=3:INSNM,IBMSORT=4:PLNTYP,IBMSORT=5:-APPT,1:PTNM)  .. S ^TMP($J,RTN,SORT,PTNM,DFN,A)=SSN\_U\_INSNM\_U\_PLNTYP\_U\_APDTE  .. Q  . Q  ;  I '$G(ZTSTOP) D PRINT             ; print the report  D ^%ZISC ; close the device  K ^TMP($J,RTN),^("IBCEPT"),^("IBSDNEXT"),^("IBDPT"),^("IBLAST") ;cleanup  I $D(ZTQUEUED) S ZTREQ="@"        ; purge the task record COMPX ;  Q  ; PRINT ; print the report to the device specified  N MAXCNT,CRT,PAGECNT,STOP,SORT,PTNM,DFN,A,DATA,DIR,X,Y,DIRUT,DIROUT,IBX  I IOST["C-" S MAXCNT=IOSL-3,CRT=1  E  S MAXCNT=IOSL-6,CRT=0  S PAGECNT=0,STOP=0  ;  ; Check for no data  I '$D(^TMP($J,RTN)) D HEADER W !!?5,"No Data Found"  ;  S SORT=""  F  S SORT=$O(^TMP($J,RTN,SORT)) Q:SORT=""  D  Q:STOP  . S PTNM=""  . F  S PTNM=$O(^TMP($J,RTN,SORT,PTNM)) Q:PTNM=""  D  Q:STOP  .. S DFN=0  .. F  S DFN=$O(^TMP($J,RTN,SORT,PTNM,DFN)) Q:'DFN  D  Q:STOP  ... S A=0  ... F  S A=$O(^TMP($J,RTN,SORT,PTNM,DFN,A)) Q:'A  D  Q:STOP  .... S DATA=$G(^TMP($J,RTN,SORT,PTNM,DFN,A))  .... I $Y+1>MAXCNT!'PAGECNT D HEADER Q:STOP  .... W !,$E(PTNM,1,20),?23,$P(DATA,U,1),?30,$E($P(DATA,U,2),1,20)  .... W ?53,$E($P(DATA,U,3),1,13),?69,$P(DATA,U,4)  .... Q  ... Q  .. Q  . Q  ;  I STOP G PRINTX  W !!?30,"\*\*\* End of Report \*\*\*"  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; HEADER ; page break and report header information  NEW LIN,HDR,TAB  S STOP=0  ; ask screen user if they want to continue  I CRT,PAGECNT>0,'$D(ZTQUEUED) D  I STOP G HEADERX  . I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S STOP=1 Q  . Q  ;  S PAGECNT=PAGECNT+1  W @IOF,!,"Patients Without MEDICARE (WNR) Insurance"  S HDR="Page: "\_PAGECNT  S TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,"Sorted by ",$S(IBMSORT=1:"Patient Name",IBMSORT=2:"SSN - Last 4 Digits",IBMSORT=3:"Insurance Company",IBMSORT=4:"Type of Plan",IBMSORT=5:"Appointment Date",1:"Patient Name")  S HDR=$$FMTE^XLFDT($$NOW^XLFDT,"1Z")  S TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,"Patient Name",?24,"SSN",?30,"Insurance Company"  W ?53,"Type of Plan",?69,"ApptDate"  W !,$$RJ^XLFSTR("",80,"=")  ;  ; check for stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HEADERX  . S (ZTSTOP,STOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HEADERX ;  Q  ; PTINS(DFN,MCRSUP) ; Function to determine if a patient should be   ; included in this report or not.  ; Input: DFN - patient ien  ; Output: Function value is either 0 (don't include) or 1 (include)  ; MCRSUP array pass by reference  ; MCRSUP(seq) = [1] insurance co ien pointer to file 36  ; [2] group pointer to file 355.3  ; [3] type of plan pointer to file 355.1  ;  NEW INCLUDE,INS,A,MCRWNR,MCRZ,IBINS,IBGRP,GP,TP,PLABBR  S INCLUDE=0 KILL MCRSUP  I '$G(DFN) G PTINSX  I '$D(^DPT(DFN)) G PTINSX  D ALLWNR^IBCNS1(DFN,"INS",DT)  S A=0,(MCRWNR,MCRZ)=0  F  S A=$O(INS(A)) Q:'A  D  Q:MCRWNR  . S IBINS=$P($G(INS(A,0)),U,1)  . S IBGRP=$P($G(INS(A,0)),U,18)  . I $$MCRWNR^IBEFUNC(IBINS) S MCRWNR=1 Q      ; Medicare WNR on file  . S GP=$G(INS(A,355.3)) ; group/plan info  . S TP=$P(GP,U,9),PLABBR=""                   ; type of plan pointer  . I TP S PLABBR=$P($G(^IBE(355.1,TP,0)),U,2) ; plan abbreviation  . I '$F(".MG.MS.COUT.","."\_PLABBR\_".") Q      ; check plan  . S MCRZ=1 ; Medicare other on file  . S MCRSUP(A)=IBINS\_U\_IBGRP\_U\_TP  . Q  ;  ; If Medicare Other was found, but no Medicare WNR, then include it  I MCRZ,'MCRWNR S INCLUDE=1  ; PTINSX ;  I 'INCLUDE K MCRSUP  Q INCLUDE  ;  ; DEVICE ;**2.6.13.5** This procedure displays a warning message and prompts for the   ; device on which to print the report.  ;  NEW ZTRTN,ZTDESC,ZTSAVE,POP  W \*7,!!!?14,"\*\*\* WARNING \*\*\*"  W !?2,"This report takes a long time to compile!"  W !!?2,"The active insurance coverage for all living patients is analyzed."  W !!?2,"It is recommended that you queue this report to the background and"  W !?2,"run it after hours or on the weekend."  W !!?2,"This report is 80 characters wide."  W !  ;  S ZTRTN="COMPILE^IBCEMMR"  S ZTDESC="Patients without MEDICARE (WNR) Insurance"  S ZTSAVE("IBMSORT")=""  D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE) DEVICEX ;  Q  ; |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCEMMR ;ALB/ESG - IB MRA Report of Patients w/o Medicare WNR ;20-NOV-2003  ;;2.0;INTEGRATED BILLING;\*\*155,366\*\*;21-MAR-94;Build 3  ;  ; Find patients with Medicare supplemental insurance or Medigap  ; insurance (etc.) but who do not have MEDICARE (WNR) on file as  ; one of their insurances.  ;  Q  ; EN ; Entry Point **; ENDDATE – End date for new filtering criteria (RSD 2.6.13.1)**  **; STARTDT – Start date for new filtering criteria**  NEW **ENDDATE**,IBMSORT,**STARTDT** **; RSD 2.6.13.1 – For use with new filtering criteria**  **D FILTER(.STARTDT,.ENDDATE)**  **I STARTDT=”” D EX Q**   NEW IBMSORT  D SORT I 'IBMSORT G EX  D DEVICE EX ; Exit Point  Q  ; **FILTER(STARTDT,ENDDATE) ; Ask users to provide start/end dates to**  **; filter by Last Appointment Date**  **; (RSD 2.6.13.1, 2.6.13.2, 2.6.13.3, 2.6.13.8, and 2.6.13.9)**  **; Logic to get start/end dates**  **; Include logic to ask/require both dates, allow default for**  **; end date, apply business rules and provide help text**  **; No future dates and allow start date = end date**  **Q**  ;  SORT ; Ask user how to sort the report **(Secondary sort) (RSD 2.6.13.1)**  NEW CH,DIR,X,Y,DIRUT,DIROUT  W @IOF,!?20,"Patients Without MEDICARE (WNR) Insurance"  W !!?2,"This option finds patients who do not have active MEDICARE (WNR) insurance,"  W !?2,"but who do have active insurance with a Plan Type of Medigap, Carve-Out, or"  W !?2,"Medicare Secondary. In these cases, MEDICARE (WNR) should be primary." **; RSD 2.6.13.8**   W !!?2,"The insurances for all living patients will be analyzed, but"  W !?2,"you can determine how this information will be sorted." S IBMSORT=""  W !  S CH="1:Patient Name;2:SSN - Last 4 Digits;3:Insurance Company;"  S CH=CH\_"4:Type of Plan**~~;5:Appointment Date~~**"  S DIR(0)="SO^"\_CH  S DIR("A")="Please enter the Sort Criteria"  S DIR("B")="Patient Name"  D ^DIR K DIR  I 'Y G SORTX  S IBMSORT=Y SORTX ;  Q  ; COMPILE ; Entry point for both background and foreground task execution  ;  **; LSTVERDT – Last verified date (RSD 2.6.13.4)**  NEW RTN,DFN,CNT,MS,DPT,PTNM,SSN,APPT,APDTE,A  NEW INS,GRP,PLN,INSNM,PLNTYP,SORT,X,IBNEXT**,LSTVERDT**  S RTN="IBCEMMR"  K ^TMP($J,RTN),^("IBCEPT"),^("IBSDNEXT"),^("IBDPT"),^("IBLAST")  S DFN=" ",CNT=0  F  S DFN=$O(^DPT(DFN),-1) Q:'DFN!($G(ZTSTOP)) D  . S CNT=CNT+1  . I '$D(ZTQUEUED),CNT#500=0 U IO(0) W "." U IO  . I $D(ZTQUEUED),CNT#500=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . I $P($G(^DPT(DFN,.35)),U,1) Q           ; date of death  . I '$$PTINS(DFN,.MS) Q                   ; eligible for report  . S ^TMP($J,"IBNEXT",DFN)=""  . S ^TMP($J,"IBLAST",DFN)=""  . S ^TMP($J,"IBDPT",DFN)=""  ;  S X=$$NEXT^IBSDU("^TMP($J,""IBNEXT"",")  S X=$$LAST^IBSDU("^TMP($J,""IBLAST"",")  ;  S DFN=0 F  S DFN=$O(^TMP($J,"IBDPT",DFN)) Q:'DFN!($G(ZTSTOP)) D  . I '$D(ZTQUEUED),CNT#500=0 U IO(0) W "." U IO  . I $D(ZTQUEUED),CNT#500=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  . I '$$PTINS(DFN,.MS) ; get MS data  . S DPT=$G(^DPT(DFN,0))  . S PTNM=$P(DPT,U,1)  . I PTNM="" S PTNM="~UNKNOWN"  . S SSN=$E($P(DPT,U,9),6,99)\_" "  . I SSN="" S SSN="~UNK"  . S (APPT,IBNEXT)=$G(^TMP($J,"IBNEXT",DFN),"UNKNOWN")  . I 'APPT S APPT=$G(^TMP($J,"IBLAST",DFN),"UNKNOWN")  . S APDTE=$S(APPT:$$FMTE^XLFDT($P(APPT,"."),"2Z"),$L(IBNEXT):IBNEXT,$L(APPT):APPT,1:"N/A") **. ; RSD 2.6.13.1**  **. ; IS THIS THE LAST APPOINTMENT DATE?**  **. ; IF THE APPOINTMENT DATE IS < STARTDT OR > ENDDT QUIT**  **. ; RSD 2.6.13.4**  **. ; DETERMINE LAST VERIFIED DATE**  **. S LSTVERDT=[Code to come]**  .  .   . S APPT=+APPT  . S A=0 F  S A=$O(MS(A)) Q:'A  D  .. S INS=+$P(MS(A),U,1),GRP=+$P(MS(A),U,2)  .. S PLN=+$P(MS(A),U,3)  .. S INSNM=$P($G(^DIC(36,INS,0)),U,1)  .. I INSNM="" S INSNM="~UNKNOWN"  .. S PLNTYP=$P($G(^IBE(355.1,PLN,0)),U,1)  .. I PLNTYP="" S PLNTYP="~UNKNOWN"  ..;  S SORT=$S(IBMSORT=1:PTNM,IBMSORT=2:SSN,IBMSORT=3:INSNM,IBMSORT=4:PLNTYP,IBMSORT=5:-APPT,1:PTNM) **.. ; Primary sort order is by Last Appointment Date with the most recent date at the top (RSD 2.6.13.7)**   .. S ^TMP($J,RTN,-**APPTDE**,PTNM,DFN,A)=SSN\_U\_INSNM\_U\_PLNTYP\_U\_APDTE\_**U\_LSTVERDT**  .. Q  . Q  ;  I '$G(ZTSTOP) D PRINT             ; print the report  D ^%ZISC ; close the device  K ^TMP($J,RTN),^("IBCEPT"),^("IBSDNEXT"),^("IBDPT"),^("IBLAST") ;cleanup  I $D(ZTQUEUED) S ZTREQ="@"        ; purge the task record COMPX ;  Q  ; PRINT ; print the report to the device specified **; APDTE – Last appointment (the primary sort criteria – RSD 2.6.13.7)**   N MAXCNT,CRT,PAGECNT,STOP,SORT,PTNM,DFN,A,DATA,DIR,X,Y,DIRUT,DIROUT,IBX**,APDTE**  **S APDTE=””**  **F S APDTE=$O(^TMP($J,RTN,APDTE)) Q:APDTE=”” D Q:STOP .**  I IOST["C-" S MAXCNT=IOSL-3,CRT=1 **.**  E  S MAXCNT=IOSL-6,CRT=0 **.**  S PAGECNT=0,STOP=0 **.**  ; **.**  ; Check for no data **.**  I '$D(^TMP($J,RTN)) D HEADER W !!?5,"No Data Found" **.**  ; **.**  S SORT="" **.**  F  S SORT=$O(^TMP($J,RTN,**APDTE,**SORT)) Q:SORT=""  D  Q:STOP **.** . S PTNM="" **.** . F  S PTNM=$O(^TMP($J,RTN,**APDTE,**SORT,PTNM)) Q:PTNM=""  D  Q:STOP **.** .. S DFN=0 **.** .. F  S DFN=$O(^TMP($J,RTN,**APDTE,**SORT,PTNM,DFN)) Q:'DFN  D  Q:STOP **.** ... S A=0 **.**... F  S A=$O(^TMP($J,RTN,**APDTE,**SORT,PTNM,DFN,A)) Q:'A  D  Q:STOP **.**.... S DATA=$G(^TMP($J,RTN,**APDTE,**SORT,PTNM,DFN,A))  **.**.... I $Y+1>MAXCNT!'PAGECNT D HEADER Q:STOP **..... ; Enable printing to delimited file (RSD 2.6.13.6)**  **..... ; A new field (Last Verified Date) is being added so a number of fields have been shortened (RSD 2.6.13.4)**  **.**.... W !,$E(PTNM,1,**16**),?**19**,$P(DATA,U,1),?**26**,$E($P(DATA,U,2),1,**16**)  **.**.... W ?**45**,$E($P(DATA,U,3),1,**9**),?**57**,$P(DATA,U,4)**,?69,$P(DATA,U,5)**  **.**.... Q  **.**... Q  **.**.. Q  **.**. Q  **.Q**   ;  I STOP G PRINTX  W !!?30,"\*\*\* End of Report \*\*\*"  I CRT,'$D(ZTQUEUED) S DIR(0)="E" D ^DIR K DIR PRINTX ;  Q  ; HEADER ; page break and report header information  NEW LIN,HDR,TAB  S STOP=0  ; ask screen user if they want to continue  I CRT,PAGECNT>0,'$D(ZTQUEUED) D  I STOP G HEADERX  . I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  . S DIR(0)="E" D ^DIR K DIR  . I 'Y S STOP=1 Q  . Q  ;  S PAGECNT=PAGECNT+1  W @IOF,!,"Patients Without MEDICARE (WNR) Insurance"  S HDR="Page: "\_PAGECNT  S TAB=80-$L(HDR)-1  W ?TAB,HDR  W !,"Sorted by ",$S(IBMSORT=1:"Patient Name",IBMSORT=2:"SSN - Last 4 Digits",IBMSORT=3:"Insurance Company",IBMSORT=4:"Type of Plan",IBMSORT=5:"Appointment Date",1:"Patient Name")  S HDR=$$FMTE^XLFDT($$NOW^XLFDT,"1Z")  S TAB=80-$L(HDR)-1 **; Enable printing to delimited file (RSD 2.6.13.6)**   W ?TAB,HDR  W !,"Patient Name",**?20**,"SSN",**?26**,"Insurance Company" **; A new field (Last Verified Date) is being added (RSD 2.6.13.4)**   W ?**45**,"Type of Plan",?**57**,"ApptDate"**,?69,”LstVerDt”**   W !,$$RJ^XLFSTR("",80,"=")  ;  ; check for stop request  I $D(ZTQUEUED),$$S^%ZTLOAD() D  G HEADERX  . S (ZTSTOP,STOP)=1  . W !!!?5,"\*\*\* Report Halted by TaskManager Request \*\*\*"  . Q  ; HEADERX ;  Q  ; PTINS(DFN,MCRSUP) ; Function to determine if a patient should be   ; included in this report or not.  ; Input: DFN - patient ien  ; Output: Function value is either 0 (don't include) or 1 (include)  ; MCRSUP array pass by reference  ; MCRSUP(seq) = [1] insurance co ien pointer to file 36  ; [2] group pointer to file 355.3  ; [3] type of plan pointer to file 355.1  ;  NEW INCLUDE,INS,A,MCRWNR,MCRZ,IBINS,IBGRP,GP,TP,PLABBR  S INCLUDE=0 KILL MCRSUP  I '$G(DFN) G PTINSX  I '$D(^DPT(DFN)) G PTINSX  D ALLWNR^IBCNS1(DFN,"INS",DT)  S A=0,(MCRWNR,MCRZ)=0  F  S A=$O(INS(A)) Q:'A  D  Q:MCRWNR  . S IBINS=$P($G(INS(A,0)),U,1)  . S IBGRP=$P($G(INS(A,0)),U,18)  . I $$MCRWNR^IBEFUNC(IBINS) S MCRWNR=1 Q      ; Medicare WNR on file  . S GP=$G(INS(A,355.3)) ; group/plan info  . S TP=$P(GP,U,9),PLABBR=""                   ; type of plan pointer  . I TP S PLABBR=$P($G(^IBE(355.1,TP,0)),U,2) ; plan abbreviation  . I '$F(".MG.MS.COUT.","."\_PLABBR\_".") Q      ; check plan  . S MCRZ=1 ; Medicare other on file  . S MCRSUP(A)=IBINS\_U\_IBGRP\_U\_TP  . Q  ;  ; If Medicare Other was found, but no Medicare WNR, then include it  I MCRZ,'MCRWNR S INCLUDE=1  ; PTINSX ;  I 'INCLUDE K MCRSUP  Q INCLUDE  ;  ; DEVICE(STARTDT,ENDDATE) ; This procedure displays a warning message and prompts for the   ; device on which to print the report.  **; Implementing a new sort order of Last Appointment Date which will accept such dates from STARTDT to ENDDATE inclusively (RSD 2.6.13.1)**   ;  NEW ZTRTN,ZTDESC,ZTSAVE,POP  W \*7,!!!?14,"\*\*\* WARNING \*\*\*"  W !?2,"This report takes a long time to compile!"  W !!?2,"The active insurance coverage for all living patients is analyzed."  W !!?2,"It is recommended that you queue this report to the background and"  W !?2,"run it after hours or on the weekend."  W !!?2,"This report is 80 characters wide."  W !  ;  S ZTRTN="COMPILE^IBCEMMR"  S ZTDESC="Patients without MEDICARE (WNR) Insurance"  S ZTSAVE("IBMSORT")="" **; ENABLE THE REPORT TO CHOOSE A FILE IN DELIMITED FORMAT (RSD 2.6.13.6)**   D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE) DEVICEX ;  Q  ; |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOMA** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.14.1, 2.6.14.2, 2.6.14.12 | | | | | | | | |
| **Related Options** | Active Policies with no Effective Date Report [IBCN POL W/NO EFF DATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | BEG^IBCOMA1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOMA ;ALB/CMS - IDENTIFY ACTIVE POLICIES W/NO EFFECTIVE DATE; 08-03-98  ;;2.0;INTEGRATED BILLING;\*\*103\*\*;21-MAR-94  Q EN ;Entry point from option  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  N IBAIB,IBBDT,IBEDT,IBRF,IBRL,IBQUIT,IBSIN,IBSTR,X,Y  S (IBAIB,IBBDT,IBEDT,IBRF,IBRL,IBSIN,IBSTR)=""  W !!,?10,"Identify Active Policies with NO Effective Date",!  S DIR("A",1)="Sort report by"  S DIR("A",2)=" 1 - Patient Name Range"  S DIR("A",3)=" 2 - Terminal Digit Range"  S DIR("A",4)=" "  S DIR(0)="SAXB^1:Patient Name;2:Terminal Digit"  S DIR("A")=" Select Number: ",DIR("B")="1",DIR("??")="^D ENH^IBCOMA" D ^DIR  I +Y'>0 S IBQUIT=1 G EXIT  S IBAIB=+Y  K DIR,DIROUT,DTOUT,DUOUT,DIRUT  W !! D @$S(IBAIB=1:"NR",1:"TR")  I $G(IBQUIT)=1 G EXIT  ; VER W !!  S DIR("A",1)=" Within "\_$S(IBAIB=1:"Patient Name",1:"Terminal Digit")\_" Include:"  S DIR("A",2)=" 1 - Verified Policies"  S DIR("A",3)=" 2 - Non-Verified Policies"  S DIR("A",4)=" 3 - Both"  S DIR("A",5)=" "  S DIR(0)="SAXB^1:Verified Policies;2:Non-Verified Policies;3:Both"  S DIR("A")=" Select Number: ",DIR("B")="1",DIR("??")="^D ICH^IBCOMA" D ^DIR  I +Y'>0 S IBQUIT=1 G EXIT  S IBSIN=+Y  K DIR,DIROUT,DTOUT,DUOUT,DIRUT  I IBSIN'=2 D VR I IBBDT=""!(IBEDT="") W " <Date Range not entered>" G VER  I $G(IBQUIT)=1 G EXIT  ;  W !! D QUE  ; EXIT Q  ; NR ; Ask Name Range  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y NRR S DIR(0)="FO",DIR("B")="FIRST",DIR("A")=" START WITH PATIENT NAME"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S:Y="FIRST" Y="A" S IBRF=Y  S DIR(0)="FO",DIR("B")="LAST",DIR("A")=" GO TO PATIENT NAME"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S:Y="LAST" Y="zzzzzz" S IBRL=Y  I $G(IBRL)']$G(IBRF) W !!,?5,"\* The Go to Patient Name must follow after the Start with Name. \*",! G NRR  Q  ; TR ; Ask Terminal Digit Range  N DIR,DIRUT,DUOUT,DTOUT,X,Y  S DIR(0)="FO^1:9^K:X'?1.9N X"  S DIR("?")="Enter up to 9 digits of the Terminal Digit to include in Report"  S DIR("B")="0000",DIR("A")=" Start with Terminal Digit"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S IBRF=$E((Y\_"000000000"),1,9)  S DIR("B")="9999",DIR("A")=" GO to Terminal Digit"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S IBRL=$E((Y\_"999999999"),1,9)  I IBRF>IBRL W !!,?5,"\* The Go to Terminal Digit must follow after the Start with Digit. \*",! G TR  Q  ; VR ; Ask Verification Date Range  N %DT,X,Y  W !!," Please enter Policy Verification Dates:"  ; VRBDT ; - get begin date  S (IBBDT,IBEDT)=""  S %DT="AEX",%DT("A")=" Start with DATE: " D ^%DT K %DT G VRQ:Y<0 S IBBDT=Y  ; VREDT ; - get ending date  S %DT="EX" R !," Go to DATE: ",X:DTIME S:X=" " X=IBBDT G VRQ:(X="")!(X["^") D ^%DT G VREDT:Y<0 S IBEDT=Y I Y<IBBDT W \*7," ??",!,"ENDING DATE must follow BEGINNING DATE." G VRBDT  ; VRQ Q  ; ENH ; Sort help Text  W !!,?5,"Enter 1 to search by a Patient Name Range. (i.e. ADAMS to ADAMSZ)"  W !!,?5,"Enter 2 to search by Terminal Digit. The output will be sorted"  W !?5,"by the 8th and 9th digits and then the 6th and 7th digits"  W !?5,"of the Patient's SSN.",!  Q  ; ICH ; Search criteria help Text  W !!,?5,"Enter 1 to list active policies by Verification Date Range"  W !,?15,"(i.e. Sort Date By: 10-1-96 Go to Date: 01-1-97)"  W !,?5,"Enter 2 to list active policies with no Verification Date."  W !,?5,"Enter 3 to include active policies with or without a Verification Date."  Q QUE ; Ask Device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  W !,?10,"You may want to queue this report!",!  S %ZIS="QM" D ^%ZIS G:POP QUEQ  I $D(IO("Q")) K IO("Q") D  G QUEQ  .S ZTRTN="BEG^IBCOMA1",ZTSAVE("IBRF")="",ZTSAVE("IBRL")=""  .S ZTSAVE("IBAIB")="",ZTSAVE("IBBDT")="",ZTSAVE("IBEDT")="",ZTSAVE("IBSIN")=""  .S ZTDESC="IB - Identify Active Policies w/no Effective Date"  .D ^%ZTLOAD K ZTSK D HOME^%ZIS  ;  U IO  I $E(IOST,1,2)["C-" W !!,?15,"... One Moment Please ..."  D BEG^IBCOMA1  ; QUEQ ; EXIT CLEAN-UP  W ! D ^%ZISC K IBAIB,IBRF,IBRL,IBSIN,IBSTR,^TMP("IBCOMA",$J)  Q  ;IBCOMA |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOMA ;ALB/CMS - IDENTIFY ACTIVE POLICIES W/NO EFFECTIVE DATE; 08-03-98  ;;2.0;INTEGRATED BILLING;\*\*103\*\*;21-MAR-94  Q EN ;Entry point from option  N DIR,DIROUT,DIRUT,DTOUT,DUOUT  **; 2.6.14.1 – Add variable IBPTYPE for Living/Deceased Patient filter**  **; 2.6.14.2 – Add variable IBAPPTE,IBAPPTS Appointment Date Range filter**  N IBAIB,IBAPPTE,IBAPPTS,IBBDT,IBEDT,**IBPTYPE**,IBRF,IBRL,IBQUIT,IBSIN,IBSTR,X,Y  S (IBAIB,IBBDT,IBEDT,IBRF,IBRL,IBSIN,IBSTR)=""  W !!,?10,"Identify Active Policies with NO Effective Date",!  S DIR("A",1)="Sort report by"  S DIR("A",2)=" 1 - Patient Name Range"  S DIR("A",3)=" 2 - Terminal Digit Range"  S DIR("A",4)=" "  S DIR(0)="SAXB^1:Patient Name;2:Terminal Digit"  S DIR("A")=" Select Number: ",DIR("B")="1",DIR("??")="^D ENH^IBCOMA" D ^DIR  I +Y'>0 S IBQUIT=1 G EXIT  S IBAIB=+Y  **; 2.6.14.1 – Add method to prompt for Living/Deceased Patient filter** **S IBPTYPE=$$PATDORA()  I IBPTYPE=-1 D EXIT Q**  K DIR,DIROUT,DTOUT,DUOUT,DIRUT  W !! D @$S(IBAIB=1:"NR",1:"TR")  I $G(IBQUIT)=1 G EXIT  ; VER W !!  S DIR("A",1)=" Within "\_$S(IBAIB=1:"Patient Name",1:"Terminal Digit")\_" Include:"  S DIR("A",2)=" 1 - Verified Policies"  S DIR("A",3)=" 2 - Non-Verified Policies"  S DIR("A",4)=" 3 - Both"  S DIR("A",5)=" "  S DIR(0)="SAXB^1:Verified Policies;2:Non-Verified Policies;3:Both"  S DIR("A")=" Select Number: ",DIR("B")="1",DIR("??")="^D ICH^IBCOMA" D ^DIR  I +Y'>0 S IBQUIT=1 G EXIT  S IBSIN=+Y  K DIR,DIROUT,DTOUT,DUOUT,DIRUT  I IBSIN'=2 D VR I IBBDT=""!(IBEDT="") W " <Date Range not entered>" G VER  I $G(IBQUIT)=1 G EXIT  ;  **; 2.6.14.2 – Add method to prompt for Appointment Date Range** **S IBAPPTS=$$APPTDR()  I IBAPPTS=-1 D EXIT Q**  **S IBAPPTE=$$APPTDR(IBAPPTS)  I IBAPPTE=-1 D EXIT Q**   W !! D QUE  ; EXIT Q  ;  **; 2.6.14.1 Prompt the user for living or deceased method PATDORA() ; Prompt the user to only display Living patients, Deceased Patients  ; or both  ; Input: None  ; Output: None  ; Returns: 1 - Living Patients Only  ; 2 - Deceased Patients only  ; 3 - Both Living and Deceased Patients  ; -1 - Not answered  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DIR("A",1)="Within the selected Range, Include:"  S DIR("A",2)=" 1 - Living Patients"  S DIR("A",3)=" 2 - Deceased Patients"  S DIR("A",3)=" 3 - Both"  S DIR("A",4)=" "  S DIR(0)="SAXB^1:Living Patients;2:Deceased Patients;3:Both"  S DIR("A")=" Select Number: ",DIR("B")="1",DIR("??")="^D PDORAH^IBCOMA"  D ^DIR  Q:+Y'>0 -1 ; Nothing selected  Q Y**  ;  **; 2.6.14.2 Prompt the user for Appointment Date Range method** APPTDR(WHICH,DTMIN) ; Prompt the user to only display Appointments in the specified range  ; Input: WHICH - 0 - Ask Start Date of the range  ; 1 - Ask End Date of the range   ; DTMIN – Minimum date that can be entered (if passed)  ; Output: None  ; Returns: Selected Date in Fileman format or -1 if not selected  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,PROMPT,TEXT,X,Y  S:'$D(DTMIN) DTMIN=""  I WHICH=0 D  . S TEXT=”Start”  . W !!,"Please enter Last Appointment Dates:"  . S PROMPT="Start with DATE"  E   S TEXT=”End”,PROMPT="Go to DATE"  S DIR("A",1)=PROMPT  S DIR(0)="DO^"\_DTMIN\_":EX"  S DIR("A")=PROMPT\_": ",DIR("B")="1",DIR("??")="^D APPTDRH^IBCOMA(TEXT)"  D ^DIR  Q:+Y'>0 -1 ; Nothing selected  Q Y  ; NR ; Ask Name Range  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y NRR S DIR(0)="FO",DIR("B")="FIRST",DIR("A")=" START WITH PATIENT NAME"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S:Y="FIRST" Y="A" S IBRF=Y  S DIR(0)="FO",DIR("B")="LAST",DIR("A")=" GO TO PATIENT NAME"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S:Y="LAST" Y="zzzzzz" S IBRL=Y  I $G(IBRL)']$G(IBRF) W !!,?5,"\* The Go to Patient Name must follow after the Start with Name. \*",! G NRR  Q  ; TR ; Ask Terminal Digit Range  N DIR,DIRUT,DUOUT,DTOUT,X,Y  S DIR(0)="FO^1:9^K:X'?1.9N X"  S DIR("?")="Enter up to 9 digits of the Terminal Digit to include in Report"  S DIR("B")="0000",DIR("A")=" Start with Terminal Digit"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S IBRF=$E((Y\_"000000000"),1,9)  S DIR("B")="9999",DIR("A")=" GO to Terminal Digit"  D ^DIR I ($D(DTOUT))!($D(DUOUT)) S IBQUIT=1 Q  S IBRL=$E((Y\_"999999999"),1,9)  I IBRF>IBRL W !!,?5,"\* The Go to Terminal Digit must follow after the Start with Digit. \*",! G TR  Q  ; VR ; Ask Verification Date Range  N %DT,X,Y  W !!," Please enter Policy Verification Dates:"  ; VRBDT ; - get begin date  S (IBBDT,IBEDT)=""  S %DT="AEX",%DT("A")=" Start with DATE: " D ^%DT K %DT G VRQ:Y<0 S IBBDT=Y  ; VREDT ; - get ending date  S %DT="EX" R !," Go to DATE: ",X:DTIME S:X=" " X=IBBDT G VRQ:(X="")!(X["^") D ^%DT G VREDT:Y<0 S IBEDT=Y I Y<IBBDT W \*7," ??",!,"ENDING DATE must follow BEGINNING DATE." G VRBDT  ; VRQ Q  ; ENH ; Sort help Text  W !!,?5,"Enter 1 to search by a Patient Name Range. (i.e. ADAMS to ADAMSZ)"  W !!,?5,"Enter 2 to search by Terminal Digit. The output will be sorted"  W !?5,"by the 8th and 9th digits and then the 6th and 7th digits"  W !?5,"of the Patient's SSN.",!  Q  ;  **; 2.6.14.12 Living or deceased filter help**  **PDORAH ; Living/Deceased/Both patient filter help Text  ; Input: NONE  ; Output: NONE  W !!,?5,"Enter 1 to only display Living Patients."  W !,?5,"Enter 2 to only display Deceased Patients."  W !,?5,"Enter 3 to only display both Living and Deceased Patients."  Q  ;**  **; 2.6.14.12 Last Appointment Date Range Help Text**  **APPTDR(WHICH) ; Last Appointment Date Range help Text  ; Input: WHICH - 0 - Start of Date Range text  ; 1 - End of Date Range text  ;   W !!,?5,"Enter a valid "\_WHICH\_" date for the last Appointment Date Range"  Q**  **;** ICH ; Search criteria help Text  W !!,?5,"Enter 1 to list active policies by Verification Date Range"  W !,?15,"(i.e. Sort Date By: 10-1-96 Go to Date: 01-1-97)"  W !,?5,"Enter 2 to list active policies with no Verification Date."  W !,?5,"Enter 3 to include active policies with or without a Verification Date."  Q QUE ; Ask Device  N %ZIS,ZTRTN,ZTSAVE,ZTDESC  W !,?10,"You may want to queue this report!",!  S %ZIS="QM" D ^%ZIS G:POP QUEQ  I $D(IO("Q")) K IO("Q") D  G QUEQ  .S ZTRTN="BEG^IBCOMA1",ZTSAVE("IBRF")="",ZTSAVE("IBRL")=""  .S ZTSAVE("IBAIB")="",ZTSAVE("IBBDT")="",ZTSAVE("IBEDT")="",ZTSAVE("IBSIN")=""  .S ZTDESC="IB - Identify Active Policies w/no Effective Date"  .D ^%ZTLOAD K ZTSK D HOME^%ZIS  ;  U IO  I $E(IOST,1,2)["C-" W !!,?15,"... One Moment Please ..."  D BEG^IBCOMA1  ; QUEQ ; EXIT CLEAN-UP  W ! D ^%ZISC K IBAIB,IBRF,IBRL,IBSIN,IBSTR,^TMP("IBCOMA",$J)  Q  ;IBCOMA |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCOMA1** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.14.1, 2.6.14.2, 2.6.14.3, 2.6.14.4, 2.6.14.5, 2.6.14.6, 2.6.14.7, 2.6.14.8, 2.6.14.9, 2.6.14.10, 2.6.14.11 | | | | | | | | |
| **Related Options** | Active Policies with no Effective Date Report [IBCN POL W/NO EFF DATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCOMA | | | | | ASK^IBCOMC2  TERMDG^IBCONS2 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCOMA1 ;ALB/CMS - IDENTIFY ACTIVE POLICIES W/NO EFFECTIVE DATE (CON'T); 08-03-98  ;;2.0;INTEGRATED BILLING;\*\*103\*\*;21-MAR-94  Q BEG ; Entry to run Active Policies w/no Effective Date Report  ; Input variables:  ; IBAIB - Required. How to sort  ; 1= Patient Name Range 2= Termianl Digit Range  ;  ; IBSIN - Required. Include Active Policies with  ; 1= Verification Date 2= No Verification Date 3= Both  ;  ; IBRF - Required. Name or Terminal Digit Range Start value  ; IBRL - Required. Name or Terminal Digit Range Go to value  ; IBBDT - Optional. Begining Verification Date Range  ; IBEDT - Optional. Ending Verification Date Range  ;  N DFN,IBC,IBC0,IBCDA,IBCDA0,IBCDA1,IBC11,IBC13,IBGP,IBI,IBPAGE,IBTMP  N IBQUIT,IBTD,IBX,VA,VADM,VAERR,X,Y  K ^TMP("IBCOMA",$J) S IBPAGE=0,IBQUIT=0  S IBC=0 F  S IBC=$O(^DPT("AB",IBC)) Q:'IBC  D  .S IBC0=$G(^DIC(36,IBC,0))  .;  .; If company inactive quit  .I $P(IBC0,U)="" Q  .I $P(IBC0,U,5)=1 Q  .S DFN=0 F  S DFN=$O(^DPT("AB",IBC,DFN)) Q:'DFN  D  ..K VA,VADM,VAERR,VAPA  ..D DEM^VADPT,ADD^VADPT  ..;  ..; I Pt. deceased quit I $G(VADM(6))>0 Q  ..; I Pt. name out of range quit  ..S VADM(1)=$P($G(VADM(1)),U,1) I VADM(1)="" Q  ..I IBAIB=1,VADM(1)]IBRL Q  ..I IBAIB=1,IBRF]VADM(1) Q  ..;  ..; I Terminal Digit out of range quit  ..I IBAIB=2 S IBTD=$$TERMDG^IBCONS2(DFN) S:IBTD="" IBTD="000000000" I (+IBTD>IBRL)!(IBRF>+IBTD) Q  ..S IBCDA=0 F  S IBCDA=$O(^DPT("AB",IBC,DFN,IBCDA)) Q:'IBCDA  D  ...S IBCDA0=$G(^DPT(DFN,.312,IBCDA,0))  ...;  ...; I Effective Date populated quit  ...I $P(IBCDA0,U,8) Q  ...;  ...; I Expiration Date entered and expired quit  ...I $P(IBCDA0,U,4),$P(IBCDA0,U,4)'>DT Q  ...;  ...; Sorting by verification date or no date check  ...S IBCDA1=$G(^DPT(DFN,.312,IBCDA,1))  ...I IBSIN=1,'$P(IBCDA1,U,3) Q  ...S $P(IBCDA1,U,3)=$P($P(IBCDA1,U,3),".",1)  ...I IBSIN=1,+$P(IBCDA1,U,3)>IBEDT Q  ...I IBSIN=1,+$P(IBCDA1,U,3)<IBBDT Q  ...I IBSIN=2,$P(IBCDA1,U,3) Q  ...I IBSIN=3 I +$P(IBCDA1,U,3)>0 I +$P(IBCDA1,U,3)<IBBDT!(+$P(IBCDA1,U,3)>IBEDT) Q  ...S IBC11=$G(^DIC(36,IBC,.11))  ...S IBC13=$G(^DIC(36,IBC,.13))  ...;  ...; set data line for global   ...;S IBTMP(1)=PT NAME^SSN^AGE^HOME PHONE^DATE OF DEATH  ...;S IBTMP(2)=INSURANCE NAME^REIMBURSE?^PHONE^ADD LINE 1  ...;S IBTMP(3)=GROUP PLAN^SUBSCRIBER ID^WHOSE INS.^VERIFICATION DATE  ...;  ...S IBGP=$P($G(^IBA(355.3,+$P(IBCDA0,U,18),0)),U,3)  ...S IBTMP(1)=VADM(1)\_U\_$P(VADM(2),U,2)\_U\_+VADM(4)\_U\_$P(VAPA(8),U,1)\_U\_$$FMTE^XLFDT($P(VADM(6),U,1),"5ZD")  ...S IBTMP(2)=$P(IBC0,U,1)\_U\_$P(IBC0,U,2)\_U\_$P(IBC13,U,1)\_U\_$P(IBC11,U,1)  ...S IBTMP(3)=$S(IBGP]"":IBGP,1:"(No Plan Name)")\_U\_$P(IBCDA0,U,2)\_U\_$P(IBCDA0,U,6)\_U\_$$FMTE^XLFDT($P(IBCDA1,U,3),"5ZD")  ...;  ...; set variable IBI for Verified=1 or Non verified=2   ...S IBI=$S(+$P(IBCDA1,U,3):1,1:2)  ...;  ...; Set Global array  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN)=IBTMP(1)  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN,IBC)=IBTMP(2)  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN,IBC,IBCDA)=IBTMP(3)  ...;  ;  I '$D(^TMP("IBCOMA",$J)) D HD W !!,"\*\* NO RECORDS FOUND \*\*" G QUEQ  D WRT  ; QUEQ ; Exit clean-UP  W ! D ^%ZISC K IBTMP,IBAIB,IBRF,IBRL,IBSIN,IBSTR,VA,VAERR,VADM,VAPA,^TMP("IBCOMA",$J)  Q  ; HD ;Write Heading  S IBPAGE=IBPAGE+1  W @IOF,"Active Policies with no Effective Date Report ",$$FMTE^XLFDT($$NOW^XLFDT,"Z")," Page: ",IBPAGE  W !,?5,"Sorted by: "\_$S(IBAIB=1:"Patient Name",1:"Terminal Digit")\_" Range: "\_$S(IBRF="A":"FIRST",1:IBRF)\_" to "\_$S(IBRL="zzzzzz":"LAST",1:IBRL)  W !,?5," Include: "\_$S(IBSIN=1:"Verification Date Range: "\_$$FMTE^XLFDT(IBBDT,"Z")\_" to "\_$$FMTE^XLFDT(IBEDT,"Z"),IBSIN=2:"No Verification Date Entered",1:"with or without Verification Date")  W !!,"Patient Name",?32,"SSN",?44,"Age",?50,"Phone",?66,"Date of Death"  W ! F IBX=1:1:79 W "="  Q  ; WRT ;Write data lines  N IBA,IBCDA,IBDA,IBDFN,IBINS,IBNA,IBPOL,IBPT,X,Y S IBQUIT=0  S IBA=0 F  S IBA=$O(^TMP("IBCOMA",$J,IBA)) Q:('IBA)!(IBQUIT=1) D  .I IBPAGE D ASK^IBCOMC2 I IBQUIT=1 Q  .D HD W !,$S(IBA=1:"Verified",1:"Non-Verified")  .S IBNA="" F  S IBNA=$O(^TMP("IBCOMA",$J,IBA,IBNA)) Q:(IBNA="")!(IBQUIT=1) D  ..S IBDFN=0 F  S IBDFN=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN)) Q:('IBDFN)!(IBQUIT=1) D  ...S IBPT=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN))  ...;  ...I ($Y+6)>IOSL D  I IBQUIT=1 Q  ....D ASK^IBCOMC2 I IBQUIT=1 Q  ....D HD  ...;  ...W !!,$E($P(IBPT,U,1),1,30),?32,$E($P(IBPT,U,2),1,12),?44,$J($P(IBPT,U,3),3),?50,$E($P(IBPT,U,4),1,20),?70,$P(IBPT,U,5)  ...;  ...S IBDA=0 F  S IBDA=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA)) Q:('IBDA)!(IBQUIT=1) D  ....S IBINS=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA))  ....W !?3,$E($P(IBINS,U,1),1,30),?35,"Reimb VA? ",$P(IBINS,U,2),?50,$E($P(IBINS,U,3),1,20) ; ?70,$E($P(IBINS,U,4),1,10)  ....;  ....S IBCDA=0 F  S IBCDA=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA,IBCDA)) Q:('IBCDA)!(IBQUIT=1) D  .....S IBPOL=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA,IBCDA))  .....W !?5,$E($P(IBPOL,U,1),1,20),?26,"Sub ID: ",$E($P(IBPOL,U,2),1,20),?55,"Whose: ",$P(IBPOL,U,3)  .....I IBA=1 W ?64,"Verif:",$P(IBPOL,U,4)  Q  ;IBCOMA1 |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCOMA1 ;ALB/CMS - IDENTIFY ACTIVE POLICIES W/NO EFFECTIVE DATE (CON'T); 08-03-98  ;;2.0;INTEGRATED BILLING;\*\*103\*\*;21-MAR-94  Q BEG ; Entry to run Active Policies w/no Effective Date Report  ; Input variables:  ; IBAIB - Required. How to sort  ; 1= Patient Name Range 2= Termianl Digit Range  ;  ; IBSIN - Required. Include Active Policies with  ; 1= Verification Date 2= No Verification Date 3= Both  ;  ; IBRF - Required. Name or Terminal Digit Range Start value  ; IBRL - Required. Name or Terminal Digit Range Go to value  ; IBBDT - Optional. Begining Verification Date Range  ; IBEDT - Optional. Ending Verification Date Range  **; 2.6.14.1, 2.6.14.2 New filter variables**  **; IBAPPTE - Last Appointment Date Range End  ; IBAPPTS - Last Appointment Date Range Start  ; IBPTYPE - Living/Deceased/Both filter ;**  N DFN,IBC,IBC0,IBCDA,IBCDA0,IBCDA1,IBC11,IBC13,IBGP,IBI,IBPAGE,IBTMP  N IBQUIT,IBTD,IBX,VA,VADM,VAERR,X,Y  K ^TMP("IBCOMA",$J) S IBPAGE=0,IBQUIT=0  S IBC=0 F  S IBC=$O(^DPT("AB",IBC)) Q:'IBC  D  .S IBC0=$G(^DIC(36,IBC,0))  .;  .; If company inactive quit  .I $P(IBC0,U)="" Q  .I $P(IBC0,U,5)=1 Q  .S DFN=0 F  S DFN=$O(^DPT("AB",IBC,DFN)) Q:'DFN  D  ..K VA,VADM,VAERR,VAPA  ..D DEM^VADPT,ADD^VADPT **..;  ..; 2.6.14.1 I Pt. deceased and not showing deceased patients quit   ..I ($G(VADM(6))>0),IBPTYPE'=2,IBPTYPE'=1 Q  ..;  ..; 2.6.14.1 I Pt. not deceased and not showing living patients quit   ..I ($G(VADM(6))'<0),IBPTYPE'=2,IBPTYPE'=0 Q**   ..;   ..; I Pt. name out of range quit  ..S VADM(1)=$P($G(VADM(1)),U,1) I VADM(1)="" Q  ..I IBAIB=1,VADM(1)]IBRL Q  ..I IBAIB=1,IBRF]VADM(1) Q  ..;  ..; I Terminal Digit out of range quit  ..I IBAIB=2 S IBTD=$$TERMDG^IBCONS2(DFN) S:IBTD="" IBTD="000000000" I (+IBTD>IBRL)!(IBRF>+IBTD) Q  ..S IBCDA=0 F  S IBCDA=$O(^DPT("AB",IBC,DFN,IBCDA)) Q:'IBCDA  D  ...S IBCDA0=$G(^DPT(DFN,.312,IBCDA,0))  ...;  ...; I Effective Date populated quit  ...I $P(IBCDA0,U,8) Q  ...;  ...; I Expiration Date entered and expired quit  ...I $P(IBCDA0,U,4),$P(IBCDA0,U,4)'>DT Q **...;  ...;2.6.14.3 If Last Appointment Date not in Last Appointment Date Range, quit**   ...;  ...; Sorting by verification date or no date check  ...S IBCDA1=$G(^DPT(DFN,.312,IBCDA,1))  ...I IBSIN=1,'$P(IBCDA1,U,3) Q  ...S $P(IBCDA1,U,3)=$P($P(IBCDA1,U,3),".",1)  ...I IBSIN=1,+$P(IBCDA1,U,3)>IBEDT Q  ...I IBSIN=1,+$P(IBCDA1,U,3)<IBBDT Q  ...I IBSIN=2,$P(IBCDA1,U,3) Q  ...I IBSIN=3 I +$P(IBCDA1,U,3)>0 I +$P(IBCDA1,U,3)<IBBDT!(+$P(IBCDA1,U,3)>IBEDT) Q  ...S IBC11=$G(^DIC(36,IBC,.11))  ...S IBC13=$G(^DIC(36,IBC,.13))  ...;  ...; set data line for global **2.6.14.11 – Add Last Appt Date**  ...;S IBTMP(1)=PT NAME^SSN^AGE^HOME PHONE^DATE OF DEATH  ...;S IBTMP(2)=INSURANCE NAME^REIMBURSE?^PHONE^ADD LINE 1  **...; 2.6.14.4 Add Last Verified By  ...;S IBTMP(3)=GROUP PLAN^SUBSCRIBER ID^WHOSE INS.^VERIFICATION DATE^VERIFIED BY^LAST APPT DATE**   ...;  ...S IBGP=$P($G(^IBA(355.3,+$P(IBCDA0,U,18),0)),U,3) **...; 2.6.14.4 Only show last 4 digits of SSN, 2.6.14.7 – 2 digit year**  **...S IBTMP(1)=VADM(1)\_U\_$P($P(VADM(2),U,2),"-",3)\_U\_+VADM(4)\_U\_$P(VAPA(8),U,1)\_U\_$$FMTE^XLFDT($P(VADM(6),U,1),"2ZD")**   ...S IBTMP(1)=VADM(1)\_U\_$P(VADM(2),U,2)\_U\_+VADM(4)\_U\_$P(VAPA(8),U,1)\_U\_$$FMTE^XLFDT($P(VADM(6),U,1),"5ZD")  ...S IBTMP(2)=$P(IBC0,U,1)\_U\_$P(IBC0,U,2)\_U\_$P(IBC13,U,1)\_U\_$P(IBC11,U,1) **...; 2.6.14.4 Add Last Verified By to line below**  **...; 2.6.14.3 Only show 2 digit year for Last Verification Date**   ...S IBTMP(3)=$S(IBGP]"":IBGP,1:"(No Plan Name)")\_U\_$P(IBCDA0,U,2)\_U\_$P(IBCDA0,U,6)\_U\_$$FMTE^XLFDT($P(IBCDA1,U,3),**"2ZD"**)  ...;  ...; set variable IBI for Verified=1 or Non verified=2   ...S IBI=$S(+$P(IBCDA1,U,3):1,1:2)  ...;  ...; Set Global array  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN)=IBTMP(1)  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN,IBC)=IBTMP(2)  ...S ^TMP("IBCOMA",$J,IBI,$S(IBAIB=2:+IBTD,1:VADM(1)),DFN,IBC,IBCDA)=IBTMP(3)  ...;  ;  I '$D(^TMP("IBCOMA",$J)) D HD W !!,"\*\* NO RECORDS FOUND \*\*" G QUEQ  D WRT  ; QUEQ ;**2.6.14.5** Exit clean-UP  **;2.6.14.6 Add ability to export to delimited file**  W ! D ^%ZISC K IBTMP,IBAIB,IBRF,IBRL,IBSIN,IBSTR,VA,VAERR,VADM,VAPA,^TMP("IBCOMA",$J)  Q  ; HD ;Write Heading  S IBPAGE=IBPAGE+1  W @IOF,"Active Policies with no Effective Date Report ",$$FMTE^XLFDT($$NOW^XLFDT,"Z")," Page: ",IBPAGE  W !,?5,"Sorted by: "\_$S(IBAIB=1:"Patient Name",1:"Terminal Digit")\_" Range: "\_$S(IBRF="A":"FIRST",1:IBRF)\_" to "\_$S(IBRL="zzzzzz":"LAST",1:IBRL)  W !,?5," Include: "\_$S(IBSIN=1:"Verification Date Range: "\_$$FMTE^XLFDT(IBBDT,"Z")\_" to "\_$$FMTE^XLFDT(IBEDT,"Z"),IBSIN=2:"No Verification Date Entered",1:"with or without Verification Date") **; 2.6.14.3, 2.6.14.9- Add Last Verified Date Column display  ; 2.6.14.4, 2.6.14.9 - Add Last Verified By Column**   W !!,"Patient Name",?32,"SSN",?44,"Age",?50,"Phone",?66,"Date of Death"  W ! F IBX=1:1:79 W "="  Q  ; WRT ;Write data lines  N IBA,IBCDA,IBDA,IBDFN,IBINS,IBNA,IBPOL,IBPT,X,Y S IBQUIT=0  S IBA=0 F  S IBA=$O(^TMP("IBCOMA",$J,IBA)) Q:('IBA)!(IBQUIT=1) D  .I IBPAGE D ASK^IBCOMC2 I IBQUIT=1 Q  .D HD W !,$S(IBA=1:"Verified",1:"Non-Verified")  .S IBNA="" F  S IBNA=$O(^TMP("IBCOMA",$J,IBA,IBNA)) Q:(IBNA="")!(IBQUIT=1) D  ..S IBDFN=0 F  S IBDFN=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN)) Q:('IBDFN)!(IBQUIT=1) D  ...S IBPT=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN))  ...;  ...I ($Y+6)>IOSL D  I IBQUIT=1 Q  ....D ASK^IBCOMC2 I IBQUIT=1 Q  ....D HD  ...;  ...W !!,$E($P(IBPT,U,1),1,30),?32,$E($P(IBPT,U,2),1,12),?44,$J($P(IBPT,U,3),3),?50,$E($P(IBPT,U,4),1,20),?70,$P(IBPT,U,5)  ...;  ...S IBDA=0 F  S IBDA=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA)) Q:('IBDA)!(IBQUIT=1) D  ....S IBINS=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA)) **....;  ....;2.6.14.10 Remove Reimb VA?**   ....W !?3,$E($P(IBINS,U,1),1,30),?35,**"Reimb VA? ",$P(IBINS,U,2)**,?50,$E($P(IBINS,U,3),1,20) ; ?70,$E($P(IBINS,U,4),1,10)  ....;  ....S IBCDA=0 F  S IBCDA=$O(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA,IBCDA)) Q:('IBCDA)!(IBQUIT=1) D  .....S IBPOL=$G(^TMP("IBCOMA",$J,IBA,IBNA,IBDFN,IBDA,IBCDA))  .....W !?5,$E($P(IBPOL,U,1),1,20),?26,"Sub ID: ",$E($P(IBPOL,U,2),1,20),?55,"Whose: ",$P(IBPOL,U,3) **....;  ....;2.6.14.4 Add Last Verified By**   .....I IBA=1 W ?64,"Verif:",$P(IBPOL,U,4) **; 2.6.14.8 - Add end of Report line  W !!,"-End of Report-"**   Q  ;IBCOMA1 |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCMDT** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.12.2, 2.6.12.3, 2.6.12.4,2.6.12.5, 2.6.12.6, 2.6.12.10 | | | | | | | | |
| **Related Options** | Missing Data Report [IBCN MISSING DATA] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | SLAI^IBCMDT1  SLAPL^IBCMDT1  SLGRN^IBCMDT1  SLPTY^IBCMDT1  SLTFT^IBCMDT1  SLEPT^IBCMDT1  SLCLM^IBCMDT1  SLBIN^IBCMDT1  SLPCN^IBCMDT1  ^IBCMDT2  FIRST^VAUTOMA  OK^IBCNSM3 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCMDT ;ALB/VD - MISSING DATA REPORT (DRIVER) ; 10-APR-15   ;;2.0;INTEGRATED BILLING ;\*\*549\*\*; 10-APR-15  ;;Per VA Directive 6402, this routine should not be modified.**  **; EN ; Describe report   W !!?5,"This report will generate a list of ACTIVE insurance companies that"  W !?5,"are missing the data that you select to be report upon.",!!  ;  ; Prompt user to select missing data to be reported.  ;  ; Output from user selections:  ; IBAI=0 -- user selects insurance companies  ; IBAI=1 -- run report for all insurance companies with plans  ; IBAPL=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies)  ; IBAPL=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ; IBGRN=0 -- ignore Missing Group Number  ; IBGRN=1 -- report Missing Group Number  ; IBPTY=0 -- ignore Missing Type of Plan  ; IBPTY=1 -- report Missing Type of Plan  ; IBTFT=0 -- ignore Missing Timely Filing Time Frame  ; IBTFT=1 -- report Missing Timely Filing Time Frame  ; IBEPT=0 -- ignore Missing Electronic Plan Type  ; IBEPT=1 -- report Missing Electronic Plan Type  ; IBCLM=0 -- ignore Missing Coverage Limitation  ; IBCLM=1 -- report Missing Coverage Limitation  ; IBBIN=0 -- ignore Missing BIN (Banking Identification Number)  ; IBBIN=1 -- report Missing BIN (Banking Identification Number)  ; IBPCN=0 -- ignore Missing PCN (Processor Control Number)  ; IBPCN=1 -- report Missing PCN (Processor Control Number)  ;  ;2.6.12.3  S IBAI=$$SLAI^IBCMDT1 I IBAI<0 G ENQ  ;2.6.12.5 - Next 8 lines  S IBAPL=$$SLAPL^IBCMDT1 I IBAPL<0 G ENQ  S IBGRN=$$SLGRN^IBCMDT1 I IBGRN<0 G ENQ  S IBPTY=$$SLPTY^IBCMDT1 I IBPTY<0 G ENQ  S IBTFT=$$SLTFT^IBCMDT1 I IBTFT<0 G ENQ  S IBEPT=$$SLEPT^IBCMDT1 I IBEPT<0 G ENQ  S IBCLM=$$SLCLM^IBCMDT1 I IBCLM<0 G ENQ  S IBBIN=$$SLBIN^IBCMDT1 I IBBIN<0 G ENQ  S IBPCN=$$SLPCN^IBCMDT1 I IBPCN<0 G ENQ**  ; **; 2.6.12.6 Add code to check to see if at least one filter was selected and quit if none selected**  **;  ; obtain plans for selected insurance companies  ;  I IBAI,IBAPL G DEVICE  D START I IBQUIT G ENQ  I '$D(^TMP("IBCMDT",$J)) W !!,"No plans selected!" G ENQ  ; DEVICE ;2.6.12.10 Ask user to select device  ;  W !!,"\*\*\* You will need a 132 column printer for this report. \*\*\*",!  S %ZIS="QM" D ^%ZIS G:POP ENQ  I $D(IO("Q")) D  G ENQ  .S ZTRTN="^IBCMDT2",ZTDESC="IB - MISSING DATA REPORT"  .F I="^TMP(""IBCMDT2"",$J,","IBAI","IBAPL" S ZTSAVE(I)=""  .D ^%ZTLOAD K IO("Q") D HOME^%ZIS  .W !!,$S($D(ZTSK):"This job has been queued as task #"\_ZTSK\_".",1:"Unable to queue this job.")  .K ZTSK,IO("Q")  ;  ; Compile and print report  ;  U IO D ^IBCMDT2  ; ENQ K DIRUT,DIROUT,DUOUT,DTOUT,IBGRNO,IBPLTY,IBTFTF,IBEPTY,IBCVLM,IBBIN,IBPCN,IBAI,IBAPL,IBQUIT,X,Y,^TMP("IBCMDT",$J)  Q  ;  ; START ; Gather plans for all selected companies.  S (IBCT,IBQUIT)=0 K ^TMP("IBCMDT",$J)  ;  ; - allow user selection of companies if required  I 'IBAI D  I Y<0 S IBQUIT=1 G STARTQ  .S DIC="^DIC(36,",DIC("S")="I $D(^IBA(355.3,""B"",Y))”  ; 2.6.12.2   .S VAUTSTR="insurance company",VAUTNI=2,VAUTVB="VAUTI",VAUTNALL=1  .D FIRST^VAUTOMA K DIC,VAUTSTR,VAUTNI,VAUTVB,VAUTNALL Q:Y<0  .S IBCNS="" F  S IBCNS=$O(VAUTI(IBCNS)) Q:IBCNS=""  S ^TMP("IBCMDT",$J,$E(VAUTI(IBCNS),1,25),IBCNS)=""  I IBAPL G STARTQ  ;  ; 2.6.12.4 - gather all companies if required  I IBAI S A=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S ^TMP("IBCMDT",$J,$E($P($G(^DIC(36,A,0)),"^"),1,25),A)=""  ;  ; - gather plans for selected companies  S IBIC="" F  S IBIC=$O(^TMP("IBCMDT",$J,IBIC)) Q:IBIC=""!IBQUIT  D  .S IBCNS="" F  S IBCNS=$O(^TMP("IBCMDT",$J,IBIC,IBCNS)) Q:IBCNS=""!(IBQUIT) D  ..S IBCT=IBCT+1 W !!,"Insurance Company # "\_IBCT\_": "\_IBIC  ..D OK^IBCNSM3 Q:IBQUIT  I 'IBOK K ^TMP("IBCMDT",$J,IBIC,IBCNS) S IBAI=0 Q  ..W " ...building a list of plans..."  ..K IBSEL,^TMP($J,"IBSEL") D LKP^IBCNSU2(IBCNS,1,1,.IBSEL,0,1) Q:IBQUIT  ..I '$O(^TMP($J,"IBSEL",0)) K ^TMP("IBCMDT",$J,IBIC,IBCNS) S IBAI=0 Q  ..;  ..; - set plans into an array  ..S IBPN=0 F  S IBPN=$O(^TMP($J,"IBSEL",IBPN)) Q:'IBPN  S ^TMP("IBCMDT",$J,IBIC,IBCNS,IBPN)=""  ; STARTQ K IBCNS,IBIC,IBJJ,IBCT,IBLCT,IBOK,IBPN,IBSEL,VAUTI,VAUTP,^TMP($J,"IBSEL")  Q** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCMDT1** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.12.3, 2.6.12.4, 2.6.12.5 | | | | | | | | |
| **Related Options** | Missing Data Report [IBCN MISSING DATA] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCMDT1 ;ALB/VD - MISSING DATA REPORT (DRIVER 1) ; 10-APR-15 ;;2.0;INTEGRATED BILLING ;\*\*549\*\*; 10-APR-15  ;;Per VA Directive 6402, this routine should not be modified.**  **; SLAI() ; 2.6.12.3 - Prompt user to select all or subset of insurance companies   ; Count ins. companies with plans  ;  ; IBV1=0 -- user selects insurance companies  ; IBV1=1 -- run report for all insurance companies with plans  ;  N A,B,IBV1  S (A,B)=0 F  S A=$O(^IBA(355.3,"B",A)) Q:'A  S B=B+1  S DIR(0)="SA^1:1. List All "\_B\_" Ins. Companies;2:2. List Only Ins. Companies That You Select"**  **; 2.6.12.4 change below code to a listman template  W !!," There are "\_B\_" insurance companies associated with plans.",!  S DIR("A",1)="1. List All "\_B\_" Ins. Companies"  S DIR("A",2)="2. List Only Ins. Companies That You Select"  S DIR("A")=" SELECT 1 or 2: "  S DIR("?",1)="Enter a code from the list: 1 or 2. Only insurance"  S DIR("?")="companies with one or more plans can be selected."  D ^DIR K DIR I Y<0!$D(DIRUT) S IBV1=-1 G SLAIQ  S IBV1=(+Y=1) K Y SLAIQ Q IBV1  ; SLAPL() ; Prompt user to select all or subset of plans  ;  ; IBV2=0 -- whether some or all ins. co's., user selects plans (may be  ; all for certain companies, some for other companies)  ; IBV2=1 -- whether some or all ins. co's., run report for all plans  ; associated with those co's.  ;  N IBV2  S DIR(0)="YO",DIR("A")="There are "\_$P(^IBA(355.3,0),"^",4)\_" plans. List all plans for each company",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will list all of the plans for each company."  S DIR("?",2)="If you say no, you must make plan selections for each individual company"  S DIR("?")="(anywhere from one plan to all)."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV2=-1 G SLAPLQ  S IBV2=+Y K Y SLAPLQ Q IBV2  ; SLGRN() ; Prompt user to report missing Group Numbers  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Group Numbers.  ; IBV3=1 -- Print missing Group Numbers.  ;  N IBV3  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Group Number",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print #'s when Group Number is missing."  S DIR("?")="If you say no, missing Group Number will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV3=-1 G SLGRNQ  S IBV3=+Y K Y SLGRNQ Q IBV3  ; SLPTY() ; Prompt user to report missing Type of Plan  ;2.6.12.5  ;  ; IBV4=0 -- Do not print missing Type of Plan.  ; IBV4=1 -- Print missing Type of Plan.  ;  N IBV4  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Type of Plan",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Type of Plan is missing."  S DIR("?")="If you say no, missing Type of Plan will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV4=-1 G SLPTYQ  S IBV4=+Y K Y SLPTYQ Q IBV4  ; SLTFT() ; Prompt user to report missing Timely Filing Time Frame  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Timely Filing Time Frame.  ; IBV3=1 -- Print missing Timely Filing Time Frame.  ;  N IBV5  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Timely Filing Time Frame",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Timely Filing Time Frame is missing."  S DIR("?")="If you say no, missing Timely Filing Time Frame will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV5=-1 G SLTFTQ  S IBV5=+Y K Y SLTFTQ Q IBV5  ; SLEPT() ; Prompt user to report missing Electronic Plan Type  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Electronic Plan Type.  ; IBV3=1 -- Print missing Electronic Plan Type.  ;  N IBV6  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Electronic Plan Type",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Electronic Plan Type is missing."  S DIR("?")="If you say no, missing Electronic Plan Type will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV6=-1 G SLEPTQ  S IBV6=+Y K Y SLEPTQ Q IBV6  ; SLCLM() ; Prompt user to report missing Coverage Limitations  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Coverage Limitations.  ; IBV3=1 -- Print missing Coverage Limitations.  ;  N IBV7  S DIR(0)="YO",DIR("A")="Do you wish to indicate missing Coverage Limitations",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Coverage Limitations is missing."  S DIR("?")="If you say no, missing Coverage Limitations will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV7=-1 G SLCLMQ  S IBV7=+Y K Y SLCLMQ Q IBV7  ; SLBIN() ; Prompt user to report missing Banking Identification Numbers (BIN)  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Banking Identification Numbers (BIN).  ; IBV3=1 -- Print missing Banking Identification Numbers (BIN).  ;  N IBV8  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Banking Identification Number (BIN)",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Banking Identification Number (BIN) is missing."  S DIR("?")="If you say no, missing Banking Identification Number (BIN) will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV8=-1 G SLBINQ  S IBV8=+Y K Y SLBINQ Q IBV8  ; SLPCN() ; Prompt user to report missing Processor Control Numbers (PCN)  ;2.6.12.5  ;  ; IBV3=0 -- Do not print missing Processor Control Numbers (PCN).  ; IBV3=1 -- Print missing Processor Control Numbers (PCN).  ;  N IBV9  S DIR(0)="YO",DIR("A")="Do you wish to indicate a missing Processor Control Number (PCN)",DIR("B")="No"  S DIR("?",1)="If you say yes, the report will print X's when Processor Control Number (PCN) is missing."  S DIR("?")="If you say no, missing Processor Control Number (PCN) will not be indicated."  W ! D ^DIR W ! K DIR I Y<0!$D(DIRUT) S IBV9=-1 G SLPCNQ  S IBV9=+Y K Y SLPCNQ Q IBV9  ;** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCMDT2** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.12.7, 2.6.12.8, 2.6.12.9 | | | | | | | | |
| **Related Options** | Missing Data Report [IBCN MISSING DATA] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCMDT2 ;ALB/VD - MISSING DATA REPORT (COMPILE) ; 10-APR-15 ;;2.0;INTEGRATED BILLING ;\*\*549\*\*; 10-APR-15  ;;Per VA Directive 6402, this routine should not be modified.**  **; EN ; Queued Entry Point for Report.  ; Required variable input: IBAI, IBAPL, IBGRN, IBPTY, IBTFT, IBEPT,  ; IBCLM, IBBIN, IBPCN  ; ^TMP("IBCMDT",$J) required if all companies and plans not selected  ;  ; - compile report data  S IBI=0 K ^TMP($J,"PR"),^TMP($J,"PL")  ;  ; - user wanted all companies and plans  I IBAI,IBAPL D  G PRINT  .S IBIC1="" F  S IBIC1=$O(^DIC(36,"B",IBIC1)) Q:IBIC1=""  D  ..S IBCNS=0 F  S IBCNS=$O(^DIC(36,"B",IBIC1,IBCNS)) Q:'IBCNS  I $D(^IBA(355.3,"B",IBCNS)) S IBIC=IBIC1 D GATH  ;  ; - user selected companies or plans  S IBIC="" F  S IBIC=$O(^TMP("IBCMDT",$J,IBIC)) Q:IBIC=""  D  .S IBCNS=0 F  S IBCNS=$O(^TMP("IBMDT",$J,IBIC,IBCNS)) Q:'IBCNS  D GATH  ; PRINT ; - print report  N EORMSG  S EORMSG="\*\*\* END OF REPORT \*\*\*" 2.6.12.9  D ^IBCMDT3  K ^TMP($J,"PR"),^TMP("IBCMDT",$J)  W !!!,EORMSG  ;  I $D(ZTQUEUED) S ZTREQ="@" Q  D ^%ZISC  K IBI,IBIC,IBIC1,IBCNS,IBCPT,IBCPS  Q  ;  ; GATH ; Gather all data for a company.  S IBI=IBI+1,(IBCPT,IBCPS,IBCST)=0 ; initialize counters  D PLAN ; gather plan info  ;  ; - set final company info  S ^TMP($J,"PR",IBI)=$$COMPINF(IBCNS)\_"^"\_IBCPT\_"^"\_IBCPS  Q  ; PLAN ; Gather Insurance Plan information, if necessary  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; initialized counters, plus the 'Plan' array (^TMP("IBINC",$J))  ;  N IBPTR,PLNDATA  S IBPTR=0 F  S IBPTR=$O(^IBA(355.3,"B",IBCNS,IBPTR)) Q:'IBPTR  D**  **.S IBCPT=IBCPT+1  .I 'IBAPL,'$D(^TMP("IBCMDT",$J,IBIC,IBCNS,IBPTR)) Q  ; not a selected plan  .S IBCPS=IBCPS+1  .S PLNDATA=$$PLANINF(IBPTR)  .I PLNDATA'=-1 S ^TMP($J,"PR",IBI,IBPTR)=PLNDATA\_"^"\_+$G(^TMP($J,"PL",IBPTR))  .D GCVLIMS(IBPTR) ; This will create the ^TMP($J,"PR",IBI,IBPTR,IBCVLM) node  K IBPTR,PLNDATA  Q  ; PLANINF(PLAN) ; Return formatted Insurance Plan information.  ; Input: PLAN -- Pointer to the plan in file #355.3  ; Output: plan number ^ name ^ grp/ind ^ act/inact  ;  N BIN,EPT,NAME,NUM,PCN,TFTF,TYP,X,X6  S X=$G(^IBA(355.3,PLAN,0)),X6=$G(^IBA(355.3,PLAN,6))  I +$P(X,"^",11) Q (-1) ; This Plan is INACTIVE, only want ACTIVE.  S NAME=$P(X,U,3) S:NAME="" NAME="#######" ;2.6.12.7  S NUM=$P(X,U,4) S:NUM="" NUM="#######" ;2.6.12.7  S TYP=$S(+$P(X,U,9):+$P($G(^IBE(355.1,+$P(X,U,9),0)),U),1:"#######")  ;2.6.12.7  S EPT=$$EXTERNAL^DILFD(355.3,.15,,PLAN)  S TFTF=$P(X,U,13) S TFTF=$S($L(TFTF):TFTF,1:"#######")  ;2.6.12.7  S BIN=$P(X6,U,2) S BIN=$S($L(BIN):BIN,1:"#######")  ;2.6.12.7  S PCN=$P(X6,U,3) S PCN=$S($L(PCN):PCN,1:"#######")  ;2.6.12.7  ; 2.6.12.8 Quit if no missing field was found**  **Q NUM\_U\_NAME\_U\_TYP\_U\_TFTF\_U\_EPT\_U\_BIN\_U\_PCN  ; COMPINF(IBCNS) ; Return formatted Insurance Company information  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; Output: company name ^ addr ^ city/st/zip ^ phone ^ precert ^ act?  ;  N ST,X,X0,X11,X13,Z  S X0=$G(^DIC(36,IBCNS,0)),X11=$G(^(.11)),X13=$G(^(.13)),Z=$P(X11,"^",6)  S ST=$S($P(X11,"^",5):$P($G(^DIC(5,$P(X11,"^",5),0)),"^",2),1:"<STATE MISSING>")  S X="Ins. Co.: "\_$E($P(X0,"^"),1,25)  S X=X\_U\_$S($P(X11,"^")'="":$P(X11,"^"),1:"<Street Addr. 1 Missing>")  S X=X\_U\_$P(X11,"^",4)\_", "\_ST\_" "\_$E(Z,1,5)\_$S($E(Z,6,9)]"":"-"\_$E(Z,6,9),1:"")  S X=X\_U\_"Phone: "\_$P(X13,"^")\_U\_"Precert Phone: "\_$P(X13,"^",3)  Q X  ; GCVLIMS(PLAN) ; Obtain Plans that may have Coverage Limits missing.  ; Input: IBCNS -- Pointer to the insurance company in file #36  ; Output: This will create the ^TMP($J,"PR",IBI,IBPTR,IBCVLM) node  ;  ; PSUEDO CODE...  ; This module will search through the Plans to obtain  ; any Coverage Limits that may be missing.  ; Code similar to FINCOV^IBCNS3 will be used.  Q  ;** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCMDT3** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.12.7 | | | | | | | | |
| **Related Options** | Missing Data Report [IBCN MISSING DATA] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT2 | | | | | DAT2^IBOUTL | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCMDT3 ;ALB/VD - MISSING DATA REPORT (PRINT) ; 10-APR-15 ;;2.0;INTEGRATED BILLING ;\*\*549\*\*; 10-APR-15  ;;Per VA Directive 6402, this routine should not be modified.**  **;  ; Print the report.  ; Required Input: Global print array ^TMP($J,"PR"  ;  N CVLMRC,CVSWT,IBC,IBCVLT,IBI,IBP,IBQUIT,IBPAG  S (IBI,IBQUIT,IBPAG)=0,IBCVLT=""  D NOW^%DTC S IBHDT=$$DAT2^IBOUTL($E(%,1,12))  F  S IBI=$O(^TMP($J,"PR",IBI)) Q:'IBI  S IBC=$G(^(IBI)) D COMP D  Q:IBQUIT  .S IBP=0 F  S IBP=$O(^TMP($J,"PR",IBI,IBP)) Q:'IBP  S IBPD=$G(^(IBP)) D  Q:IBQUIT  ..I $Y>(IOSL-5) D PAUSE Q:IBQUIT  D COMP  ..D PLAN  ..S IBCVLT=""  ..F  S IBCVLT=$O(^TMP($J,"PR",IBI,IBP,IBCVLT)) Q:IBCVLT=""  D  Q:IBQUIT  ...S CVLMRC=$G(^TMP($J,"PR",IBI,IBP,IBCVLT))  ...I +CVSWT D CVLMHD S CVSWT=0  ...W !?4,IBCVLT,?24,$P(CVLMRC,U,1),?42,$P(CVLMRC,U,2)  .Q:IBQUIT  .;  .; - print company totals  .I $Y>(IOSL-4) D PAUSE Q:IBQUIT  D COMP,PLAN  .W !!?90,"Number of Plans Selected = ",$P(IBC,"^",9)  .D PAUSE  ;  K IBJJ,IBI,IBQUIT,IBPAG,IBHDT,IBC,IBP,IBPD,IBS,IBSD,IBCVLM  Q  ; COMP ; Print Company header  I $E(IOST,1,2)="C-"!(IBPAG) W @IOF  S IBPAG=IBPAG+1  W !,"LIST OF PLANS MISSING DATA BY INSURANCE COMPANY"  W ?IOM-34,IBHDT,?IOM-10,"Page: ",IBPAG  W !,$TR($J(" ",IOM)," ","-")  ;  ; - sub-header  W !?1,$P(IBC,"^"),?45,$P(IBC,"^",4),?105,$P(IBC,"^",6)  W !!?2,"GROUP NUMBER",?27,"GROUP NAME",?40,"TYPE OF PLAN",?53,"ELEC PLAN",?59,"FTF",?65,"BIN",?72,"PCN"  Q  ; PLAN ; Print plan information.  W !!?2,$P(IBPD,U),?27,$P(IBPD,U,2),?40,$P(IBPD,U,3),?53,$P(IBPD,U,4),?59,$P(IBPD,U,5),?65,$P(IBPD,U,6),?72,$P(IBPD,U,7)  Q  ; CVLMHD ; Print Coverage Limit sub-header  ...W !?4,IBCVLT,?24,$P(CVLMRC,U,1),?42,$P(CVLMRC,U,2)  W !!?4,"Coverage",?24,"Effective Date",?42,"Covered?"  W !?4,"--------",?24,"--------------",?42,"--------"  Q  ;  ; PAUSE ; Pause for screen output.  Q:$E(IOST,1,2)'["C-"  F IBJJ=$Y:1:(IOSL-7) W !  S DIR(0)="E" D ^DIR K DIR I $D(DIRUT)!($D(DUOUT)) S IBQUIT=1 K DIRUT,DTOUT,DUOUT  Q** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNERPH | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.15.11, 2.6.15.12 | | | | | | | | |
| **Related Options** | eIV Patient Insurance Update Report [IBCNE EIV UPDATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNERP1 | | | | | FO^IBCNEUT1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNERPH ;BP/YMG - IBCNE EIV INSURANCE UPDATE REPORT PRINT;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; variables from IBCNERPF and IBCNERPG:  ; IBCNERTN = "IBCNERPF"  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  ; Summary report:  ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Count  ; SORT1 - Payer Name or \*, SORT2 - Clerk Name or 0 if not processed  ;  ; Detailed report:  ; ^TMP($J,IBCNERTN,SORT1)=Count   ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Pat. Name ^ SSN ^ Date received ^ Payer Name ^ Ck AB ^ Clerk Name ^ Date Verified ^ Days old  ; SORT1 - Payer Name, Patient Name, or Clerk Name, SORT2 - Date received  ;  Q  ; EN(IBCNERTN,IBCNESPC) ; Entry point  N CLNAME,CRT,DDATA,DLINE,EORMSG,IBPGC,IBPXT,MAXCNT,NONEMSG,NPROC,SSN,SSNLEN,SRT1,SRT2,TSTAMP,TYPE,VDATE,WIDTH,X,Y  S (IBPGC,IBPXT)=0  S NONEMSG="\* \* \* N O D A T A F O U N D \* \* \*"  S EORMSG="\*\*\* END OF REPORT \*\*\*"  S NPROC="Not Processed"  S TSTAMP=$$FMTE^XLFDT($$NOW^XLFDT,1) ; time of report  S TYPE=$G(IBCNESPC("TYPE")) ; report type  S WIDTH=$S(TYPE="S":79,1:131)  ; Determine IO parameters  S MAXCNT=IOSL-6,CRT=0  S:IOST["C-" MAXCNT=IOSL-3,CRT=1  ; print data  S SRT1=""  D HEADER I $G(ZTSTOP)!IBPXT Q  ; If global does not exist - display No Data message  I '$D(^TMP($J,IBCNERTN)) D LINE($$FO^IBCNEUT1(NONEMSG,$$CENTER(NONEMSG),"R")) G EXIT  I TYPE="S" D  Q:$G(ZTSTOP)!IBPXT  .; summary report  .F  S SRT1=$O(^TMP($J,IBCNERTN,SRT1)) Q:SRT1=""!$G(ZTSTOP)!IBPXT  D  ..I SRT1'="\*" D LINE(SRT1)  ..S SRT2="" F  S SRT2=$O(^TMP($J,IBCNERTN,SRT1,SRT2)) Q:SRT2=""!$G(ZTSTOP)!IBPXT  D  ...D LINE($$FO^IBCNEUT1(" "\_$S(SRT2=0:NPROC,1:SRT2),40)\_"Count = "\_^TMP($J,IBCNERTN,SRT1,SRT2))  ...Q  ..Q  .Q  I TYPE="D" D  Q:$G(ZTSTOP)!IBPXT  .; detailed report  .F  S SRT1=$O(^TMP($J,IBCNERTN,SRT1)) Q:SRT1=""!$G(ZTSTOP)!IBPXT  D  ..D LINE($$FO^IBCNEUT1($S(SRT1=0:NPROC,1:SRT1),85)\_"Count = "\_^TMP($J,IBCNERTN,SRT1))  ..S SRT2="" F  S SRT2=$O(^TMP($J,IBCNERTN,SRT1,SRT2)) Q:SRT2=""!$G(ZTSTOP)!IBPXT  D  ...S DDATA=$G(^TMP($J,IBCNERTN,SRT1,SRT2)),DLINE=""  ...S $E(DLINE,3,22)=$P(DDATA,U)  ...S SSN=$P(DDATA,U,2),SSNLEN=$L(SSN),$E(DLINE,23,28)=$E(SSN,SSNLEN-3,SSNLEN)  ...S $E(DLINE,29,41)=$$FMTE^XLFDT($P(DDATA,U,3)\1,"5Z")  ...S $E(DLINE,42,69)=$P(DDATA,U,4),$E(DLINE,72,77)=$P(DDATA,U,5)  ...S CLNAME=$P(DDATA,U,6) S:CLNAME=0 CLNAME=NPROC S $E(DLINE,78,94)=CLNAME  ...S VDATE=$$FMTE^XLFDT($P(DDATA,U,7)\1,"5Z") S:'VDATE VDATE=" N/A" S $E(DLINE,115,127)=VDATE  ...S $E(DLINE,127,131)=$P(DDATA,U,8)  ...D LINE(DLINE)  ...Q  ..Q  .Q  ; EXIT ;  D LINE($$FO^IBCNEUT1(EORMSG,$$CENTER(EORMSG),"R"))  I CRT,IBPGC>0,'$D(ZTQUEUED) D EOL  Q  ; EOL ; display "end of page" message and set exit flag  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,LIN  I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  S DIR(0)="E" D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) S IBPXT=1  Q  ; HEADER ; print header for each page  N DASHES,HDR,OFFSET,SRT  ;  I CRT,IBPGC>0,'$D(ZTQUEUED) D EOL I IBPXT Q  I $D(ZTQUEUED),$$S^%ZTLOAD() S (ZTSTOP,IBPXT)=1 Q  S IBPGC=IBPGC+1  W @IOF,!,?1,"Pt. Insurance Update Report"  S HDR=TSTAMP\_" Page: "\_IBPGC,OFFSET=WIDTH-$L(HDR)  W ?OFFSET,HDR  S SRT=$G(IBCNESPC("SORT"))  I TYPE="S" W !,?1,"Sorted by: Clerk Name"  I TYPE="D" W !,?1,"Sorted by: "\_$S(SRT=1:"Payer Name",SRT=2:"Patient Name",1:"Clerk Name")  S HDR=$$FMTE^XLFDT($G(IBCNESPC("BEGDT")),"5Z")\_" - "\_$$FMTE^XLFDT($G(IBCNESPC("ENDDT")),"5Z")  S OFFSET=WIDTH-$L(HDR)  W ?OFFSET,HDR  W !,?1,$S(TYPE="D":"Detailed",1:"Summary")\_" Report: "  W $S($G(IBCNESPC("PAT"))="A":"All",1:"Selected")\_" Patients; "  W $S($G(IBCNESPC("PYR"))="A":"All",1:"Selected")\_" Payers"  I TYPE="D" W !!,?3,"Patient Name",?23,"SSN",?29,"Dt Rec'd",?42,"Payer",?70,"Ck AB",?78,"Clerk/Auto",?115,"Verified",?127,"Days"  S $P(DASHES,"-",WIDTH)="" W !,?1,DASHES  Q  ; LINE(LINE) ; Print line of data  I $Y+1>MAXCNT D HEADER I $G(ZTSTOP)!IBPXT Q  W !,?1,LINE  Q  ; CENTER(LINE) ; return length of a centered line  ; LINE - line to center  N LENGTH,OFFSET  S LENGTH=$L(LINE),OFFSET=IOM-$L(LINE)\2  Q OFFSET+LENGTH |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNERPH ;BP/YMG - IBCNE EIV INSURANCE UPDATE REPORT PRINT;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; variables from IBCNERPF and IBCNERPG:  ; IBCNERTN = "IBCNERPF"  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  ; Summary report:  ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Count  ; SORT1 - Payer Name or \*, SORT2 - Clerk Name or 0 if not processed  ;  ; Detailed report:  ; ^TMP($J,IBCNERTN,SORT1)=Count   ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Pat. Name ^ SSN ^ Date received ^ Payer Name ^ Ck AB ^ Clerk Name ^ Date Verified ^ Days old  ; SORT1 - Payer Name, Patient Name, or Clerk Name, SORT2 - Date received  ;  Q  ; EN(IBCNERTN,IBCNESPC) ; Entry point  N CLNAME,CRT,DDATA,DLINE,EORMSG,IBPGC,IBPXT,MAXCNT,NONEMSG,NPROC,SSN,SSNLEN,SRT1,SRT2,TSTAMP,TYPE,VDATE,WIDTH,X,Y  S (IBPGC,IBPXT)=0  S NONEMSG="\* \* \* N O D A T A F O U N D \* \* \*"  S EORMSG="\*\*\* END OF REPORT \*\*\*"  S NPROC="Not Processed"  S TSTAMP=$$FMTE^XLFDT($$NOW^XLFDT,1) ; time of report  S TYPE=$G(IBCNESPC("TYPE")) ; report type  S WIDTH=$S(TYPE="S":79,1:131)  ; Determine IO parameters  S MAXCNT=IOSL-6,CRT=0  S:IOST["C-" MAXCNT=IOSL-3,CRT=1  ; print data  S SRT1=""  D HEADER I $G(ZTSTOP)!IBPXT Q  ; If global does not exist - display No Data message  I '$D(^TMP($J,IBCNERTN)) D LINE($$FO^IBCNEUT1(NONEMSG,$$CENTER(NONEMSG),"R")) G EXIT  I TYPE="S" D  Q:$G(ZTSTOP)!IBPXT  .; summary report  .F  S SRT1=$O(^TMP($J,IBCNERTN,SRT1)) Q:SRT1=""!$G(ZTSTOP)!IBPXT  D  ..I SRT1'="\*" D LINE(SRT1)  ..S SRT2="" F  S SRT2=$O(^TMP($J,IBCNERTN,SRT1,SRT2)) Q:SRT2=""!$G(ZTSTOP)!IBPXT  D  ...D LINE($$FO^IBCNEUT1(" "\_$S(SRT2=0:NPROC,1:SRT2),40)\_"Count = "\_^TMP($J,IBCNERTN,SRT1,SRT2))  ...Q  ..Q  .Q  I TYPE="D" D  Q:$G(ZTSTOP)!IBPXT  .; detailed report  **.; RSD 2.6.15.12 – Gather new data elements: Insurance Company, date the**  **.; eIV inquiry was sent, the date the policy was auto updated, and the eIV Trace number.**   .F  S SRT1=$O(^TMP($J,IBCNERTN,SRT1)) Q:SRT1=""!$G(ZTSTOP)!IBPXT  D  ..D LINE($$FO^IBCNEUT1($S(SRT1=0:NPROC,1:SRT1),85)\_"Count = "\_^TMP($J,IBCNERTN,SRT1))  ..S SRT2="" F  S SRT2=$O(^TMP($J,IBCNERTN,SRT1,SRT2)) Q:SRT2=""!$G(ZTSTOP)!IBPXT  D  ...S DDATA=$G(^TMP($J,IBCNERTN,SRT1,SRT2)),DLINE=""  ...S $E(DLINE,3,22)=$P(DDATA,U)  ...S SSN=$P(DDATA,U,2),SSNLEN=$L(SSN),$E(DLINE,23,28)=$E(SSN,SSNLEN-3,SSNLEN)  ...S $E(DLINE,29,41)=$$FMTE^XLFDT($P(DDATA,U,3)\1,"5Z")  ...S $E(DLINE,42,69)=$P(DDATA,U,4),$E(DLINE,72,77)=$P(DDATA,U,5)  ...S CLNAME=$P(DDATA,U,6) S:CLNAME=0 CLNAME=NPROC S $E(DLINE,78,94)=CLNAME  ...S VDATE=$$FMTE^XLFDT($P(DDATA,U,7)\1,"5Z") S:'VDATE VDATE=" N/A" S $E(DLINE,115,127)=VDATE  ...S $E(DLINE,127,131)=$P(DDATA,U,8)  ...D LINE(DLINE)  **...;2.6.15.12 Add new data elements here**   ...Q  ..Q  .Q  ; EXIT ;  D LINE($$FO^IBCNEUT1(EORMSG,$$CENTER(EORMSG),"R"))  I CRT,IBPGC>0,'$D(ZTQUEUED) D EOL  Q  ; EOL ; display "end of page" message and set exit flag  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,LIN  I MAXCNT<51 F LIN=1:1:(MAXCNT-$Y) W !  S DIR(0)="E" D ^DIR K DIR  I $D(DTOUT)!$D(DUOUT) S IBPXT=1  Q  ; HEADER ; print header for each page  N DASHES,HDR,OFFSET,SRT  ;  I CRT,IBPGC>0,'$D(ZTQUEUED) D EOL I IBPXT Q  I $D(ZTQUEUED),$$S^%ZTLOAD() S (ZTSTOP,IBPXT)=1 Q  S IBPGC=IBPGC+1  W @IOF,!,?1,"Pt. Insurance Update Report"  S HDR=TSTAMP\_" Page: "\_IBPGC,OFFSET=WIDTH-$L(HDR)  W ?OFFSET,HDR  S SRT=$G(IBCNESPC("SORT"))  I TYPE="S" W !,?1,"Sorted by: Clerk Name"  I TYPE="D" W !,?1,"Sorted by: "\_$S(SRT=1:"Payer Name",SRT=2:"Patient Name",1:"Clerk Name")  S HDR=$$FMTE^XLFDT($G(IBCNESPC("BEGDT")),"5Z")\_" - "\_$$FMTE^XLFDT($G(IBCNESPC("ENDDT")),"5Z")  S OFFSET=WIDTH-$L(HDR)  W ?OFFSET,HDR  W !,?1,$S(TYPE="D":"Detailed",1:"Summary")\_" Report: "  W $S($G(IBCNESPC("PAT"))="A":"All",1:"Selected")\_" Patients; "  W $S($G(IBCNESPC("PYR"))="A":"All",1:"Selected")\_" Payers"  **; RSD 2.6.15.11 – Remove Column headers: date the eIV response was received, ‘Ck AB’, the ‘Clerk/Auto’, and ‘Verified’.**  **; RSD 2.6.15.12 – Add Column headers: Insurance Company, date the**  **; eIV inquiry was sent, the date the policy was auto updated, and the eIV Trace number.**   I TYPE="D" W !!,?3,"Patient Name",?23,"SSN",?29,"Dt Rec'd",?42,"Payer",?70,"Ck AB",?78,"Clerk/Auto",?115,"Verified",?127,"Days"  S $P(DASHES,"-",WIDTH)="" W !,?1,DASHES  Q  ; LINE(LINE) ; Print line of data  I $Y+1>MAXCNT D HEADER I $G(ZTSTOP)!IBPXT Q  W !,?1,LINE  Q  ; CENTER(LINE) ; return length of a centered line  ; LINE - line to center  N LENGTH,OFFSET  S LENGTH=$L(LINE),OFFSET=IOM-$L(LINE)\2  Q OFFSET+LENGTH |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNERP1 | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.15.1, 2.6.15.13, 2.6.15.14 | | | | | | | | |
| **Related Options** | eIV Patient Insurance Update Report [IBCNE EIV UPDATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNBLA1  IBCNERPF  IBCNERP4  IBCNERP7 | | | | | EN^IBCNERP2  EN^IBCNERP5  EN^IBCNERP8  EN^IBCNERP9  EN3^IBCNERPA  EN6^IBCNERPA  EN^IBCNERPG  EN^IBCNERPH  FO^IBCNEUT1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNERP1 ;DAOU/BHS - IBCNE USER IF eIV RESPONSE REPORT ;03-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*184,271,416\*\*;21-MAR-94;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; eIV - Insurance Verification Interface  ;  ; Input parameters: N/A  ; Other relevant variables ZTSAVED for queueing:  ; IBCNERTN = "IBCNERP1" (current routine name for queueing the   ; COMPILE process)  ; IBCNESPC("BEGDT")=start dt for rpt  ; IBCNESPC("ENDDT")=end dt for rpt  ; IBCNESPC("PYR")=payer ien (365.12) or "" for all payers  ; IBCNESPC("SORT")=1 (Payer name) OR 2 (Patient name)  ; IBCNESPC("PAT")=patient ien (2) or "" for all patients  ; IBCNESPC("TYPE")=A (All Responses) for date range OR M (Most Recent  ; Responses) for date range (by unique Payer/Pat pair)  ; IBCNESPC("TRCN")=Trace #^IEN, if non-null all other params are null  ; IBCNESPC("RFLAG")=Report Flag used to indicate which report is being  ; run. Response Report (0), Inactive Report (1), or Ambiguous   ; Report (2).  ; IBCNESPC("DTEXP")=Expiration date used in the inactive policy report  ;  ; Only call this routine at a tag  Q EN(IPRF) ; Main entry pt  ; Init vars  N STOP,IBCNERTN,POP,IBCNESPC  S IBCNESPC("RFLAG")=$G(IPRF)  ;  S STOP=0  S IBCNERTN="IBCNERP1"  W @IOF  W !,"eIV ",$S(IPRF=1:"Inactive Policy",IPRF=2:"Ambiguous Policy",1:"Response")," Report",!  I $G(IPRF) D  . W !,"Please select a date range to view ",$S(IPRF=1:"inactive",1:"ambiguous")," policy information that the eIV"  . W !,"process turned up while attempting to discover previously unknown"  . W !,"insurance policies. (Date range selection is based on the date that"  . W !,"eIV receives the response from the payer.)"  ;  I '$G(IPRF) D  . W !,"Insurance verification responses are received daily."  . W !,"Please select a date range in which responses were received to view the"  . W !,"associated response detail. Otherwise, select a Trace # to view specific"  . W !,"response detail."  ;  ; Rpt by Date Range or Trace # R05 I '$G(IPRF) D RTYPE I STOP G:$$STOP EXIT G R05  ; If rpt by Trace # - no other criteria is necessary  I $G(IBCNESPC("TRCN")) G R100  ; Date Range params R10 D DTRANGE I STOP G:$$STOP EXIT G R05  ; Payer Selection param R20 D PYRSEL I STOP G:$$STOP EXIT G R10  ; Patient Selection param R30 D PTSEL I STOP G:$$STOP EXIT G R20  ; Type of data to return param R40 D TYPE I STOP G:$$STOP EXIT G R30  ; How far back do you want the expiration date R45 I $G(IPRF)=1 D DTEXP I STOP G:$$STOP EXIT G R40  ; Sort by param - Payer or Patient R50 D SORT I STOP G:$$STOP EXIT G R45  ; Select output device R100 D DEVICE(IBCNERTN,.IBCNESPC) I STOP G:$$STOP EXIT G:$G(IBCNESPC("TRCN"))'="" R05 G R50  G EXIT  ; EXIT ; Exit pt  Q  ;  ; COMPILE(IBCNERTN,IBCNESPC) ;   ; Entry point called from EN^XUTMDEVQ in either direct or queued mode.  ; Input params:  ; IBCNERTN = Routine name for ^TMP($J,...  ; IBCNESPC = Array passed by ref of the report params  ;  ; Init scratch globals  K ^TMP($J,IBCNERTN),^TMP($J,IBCNERTN\_"X")  ; Compile  I IBCNERTN="IBCNERP1" D EN^IBCNERP2(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERP4" D EN^IBCNERP5(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERP7" D EN^IBCNERP8(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERPF" D EN^IBCNERPG(IBCNERTN,.IBCNESPC)  ; Print  I '$G(ZTSTOP) D  . I IBCNERTN="IBCNERP1" D EN3^IBCNERPA(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERP4" D EN6^IBCNERPA(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERP7" D EN^IBCNERP9(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERPF" D EN^IBCNERPH(IBCNERTN,.IBCNESPC)  ; Close device  D ^%ZISC  ; Kill scratch globals  K ^TMP($J,IBCNERTN),^TMP($J,IBCNERTN\_"X")  ; Purge task record  I $D(ZTQUEUED) S ZTREQ="@"  ; COMPILX ; COMPILE exit pt  Q  ; STOP() ; Determine if user wants to exit out of the whole option  ; Init vars  N DIR,X,Y,DIRUT  ;  W !  S DIR(0)="Y"  S DIR("A")="Do you want to exit out of this option entirely"  S DIR("B")="YES"  S DIR("?",1)=" Enter YES to immediately exit out of this option."  S DIR("?")=" Enter NO to return to the previous question."  D ^DIR K DIR  I $D(DIRUT) S (STOP,Y)=1 G STOPX  I 'Y S STOP=0  ; STOPX ; STOP exit pt  Q Y  ; DTRANGE ; Determine start and end dates for date range param  ; Init vars  N X,Y,DIRUT  ;  W !  ;  S DIR(0)="D^:-NOW:EX"  S DIR("A")="Start DATE"  S DIR("?",1)=" Please enter a valid date for which an eIV Response"  S DIR("?")=" would have been received. Future dates are not allowed."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTRANGX  S IBCNESPC("BEGDT")=Y  ; End date DTRANG1 S DIR(0)="DA^"\_Y\_":-NOW:EX"  S DIR("A")=" End DATE: "  S DIR("?",1)=" Please enter a valid date for which an eIV Response"  S DIR("?",2)=" would have been received. This date must not precede"  S DIR("?")=" the Start Date. Future dates are not allowed."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTRANGX  S IBCNESPC("ENDDT")=Y  ; DTRANGX ; DTRANGE exit pt  Q  ; PYRSEL ; Select one payer or ALL - File #365.12  ; Init vars  NEW DIC,DTOUT,DUOUT,X,Y  ;  W !  S DIC(0)="ABEQ"  S DIC("A")=$$FO^IBCNEUT1("Payer or <Return> for All Payers: ",40,"R")  ; Do not allow selection of '~NO PAYER' and non-eIV payers  S DIC("S")="I ($P(^(0),U,1)'=""~NO PAYER""),$$PYRAPP^IBCNEUT5(""IIV"",$G(Y))'="""""  S DIC="^IBE(365.12,"  D ^DIC  I $D(DUOUT)!$D(DTOUT) S STOP=1 G PYRSELX  ; If nothing was selected (Y=-1), select ALL payers  S IBCNESPC("PYR")=$S(Y=-1:"",1:$P(Y,U,1))  ; PYRSELX ; PYRSEL exit pt  Q  ; PTSEL ; Select one patient or ALL - File #2  ; Init vars  NEW DIC,DTOUT,DUOUT,X,Y  ; Patient lookup  W !  S DIC(0)="AEQM"  S DIC("A")=$$FO^IBCNEUT1("Patient or <Return> for All Patients: ",40,"R")  S DIC="^DPT("  D ^DIC  I $D(DUOUT)!$D(DTOUT) S STOP=1 G PTSELX  ; If nothing was selected (Y=-1), select ALL patients  S IBCNESPC("PAT")=$S(Y=-1:"",1:$P(Y,U,1))  ; PTSELX ; PTSEL exit pt  Q  ; TYPE ; Prompt to select to display All or Most Recent Responses for  ; Patient/Payer combos  ; Init vars  N DIR,X,Y,DIRUT  ;  S DIR(0)="S^A:All Responses;M:Most Recent Responses"  S DIR("A")="Select the type of responses to display"  S DIR("B")="A"  S DIR("?",1)=" A - All responses from the payer during the date range will be"  S DIR("?",2)=" displayed for each unique payer/patient combination."  S DIR("?",3)=" (Default)"  S DIR("?",4)=" M - Only the most recently received response from the payer"  S DIR("?",5)=" during the date range will be displayed for each unique"  S DIR("?")=" payer/patient combination."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G TYPEX  S IBCNESPC("TYPE")=Y  ; TYPEX ; TYPE exit pt  Q  ; DTEXP ; Prompt for oldest expiration date to pull for.  ; Init Vars  N Y,DIRUT,TODAY  ;  W !  ;  S DIR(0)="D^:-NOW:EX"  S DIR("A")="Earliest Policy Expiration Date to Select From"  S DIR("B")="T-365"  S DIR("?",1)=" Please enter a valid date in the past. Any policy with a reported"  S DIR("?")=" expiration date prior to this date will not be selected."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTEXPX  S IBCNESPC("DTEXP")=Y  ; DTEXPX ; DTEXP Exit  Q  ; SORT ; Prompt to allow users to sort the report by Payer(default) or   ; Patient  ; Init vars  N DIR,X,Y,DIRUT  ;  S DIR(0)="S^1:Payer Name;2:Patient Name"  S DIR("A")="Select the primary sort field"  S DIR("B")=1  S DIR("?",1)=" 1 - Payer Name is the primary sort, Patient Name is secondary."  S DIR("?",2)=" (Default)"  S DIR("?")=" 2 - Patient Name is the primary sort, Payer Name is secondary."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G SORTX  S IBCNESPC("SORT")=Y  ; SORTX ; SORT exit pt  Q  ; RTYPE ; Prompt to allow users to report by date range or Trace #  ; Init vars  N D,DIC,DIR,X,Y,DIRUT,DTOUT,DUOUT  ;  S DIR(0)="S^1:Report by Date Range;2:Report by Trace #"  S DIR("A")="Select the type of report to generate"  S DIR("B")=1  S DIR("?",1)=" 1 - Generate report by date range, payer range, patient range"  S DIR("?",2)=" and All or Most Recent responses for payer/patient."  S DIR("?",3)=" (Default)"  S DIR("?",4)=" 2 - Generate report for a specific Trace # which corresponds"  S DIR("?")=" to an unique response."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G RTYPEX  I Y=1 S IBCNESPC("TRCN")="" G RTYPEX  ;  ; Allow user to select Trace # from x-ref "C"  W !  S DIC(0)="AEVZSQ"  S DIC="^IBCN(365,",D="C",DIC("A")="Enter Trace # for report: "  S DIC("W")="N IBX S IBX=$P($G(^(0)),U,2,3) W:$P(IBX,U,1) $P($G(^DPT($P(IBX,U,1),0)),U,1) W:$P(IBX,U,2) "" ""\_$P($G(^IBE(365.12,$P(IBX,U,2),0)),U,1)"  D IX^DIC K DIC  I $D(DTOUT)!$D(DUOUT) S STOP=1 G RTYPEX  I 'Y!(Y<0) S STOP=1 G RTYPEX  S IBCNESPC("TRCN")=$P(Y(0),U,9)\_"^"\_$P(Y,U,1)  ; RTYPEX ; RTYPE exit pt  Q  ; DEVICE(IBCNERTN,IBCNESPC) ; Device Handler and possible TaskManager calls  ;  ; Input params:  ; IBCNERTN = Routine name for ^TMP($J,...  ; IBCNESPC = Array passed by ref of the report params  ;  ; Init vars  **;2.6.15.13 Already allows print to paper**  N ZTRTN,ZTDESC,ZTSAVE,POP  ;  I IBCNERTN="IBCNERP4"!(IBCNERTN="IBCNERPF"&($G(IBCNESPC("TYPE"))="D")) W !!!,"\*\*\* This report is 132 characters wide \*\*\*",!  S ZTRTN="COMPILE^IBCNERP1("""\_IBCNERTN\_""",.IBCNESPC)"  S ZTDESC="IBCNE eIV "\_$S(IBCNERTN="IBCNERP1":"Response",IBCNERTN="IBCNERPF":"Insurance Update",1:"Payer")\_" Report"  S ZTSAVE("IBCNESPC(")=""  S ZTSAVE("IBCNERTN")=""  D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE)  I POP S STOP=1  ; DEVICEX ; DEVICE exit pt  Q  ; |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNERP1 ;DAOU/BHS - IBCNE USER IF eIV RESPONSE REPORT ;03-JUN-2002  ;;2.0;INTEGRATED BILLING;\*\*184,271,416\*\*;21-MAR-94;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; eIV - Insurance Verification Interface  ;  ; Input parameters: N/A  ; Other relevant variables ZTSAVED for queueing:  ; IBCNERTN = "IBCNERP1" (current routine name for queueing the   ; COMPILE process)  ; IBCNESPC("BEGDT")=start dt for rpt  ; IBCNESPC("ENDDT")=end dt for rpt  ; IBCNESPC("PYR")=payer ien (365.12) or "" for all payers  ; IBCNESPC("SORT")=1 (Payer name) OR 2 (Patient name)  ; IBCNESPC("PAT")=patient ien (2) or "" for all patients  ; IBCNESPC("TYPE")=A (All Responses) for date range OR M (Most Recent  ; Responses) for date range (by unique Payer/Pat pair)  ; IBCNESPC("TRCN")=Trace #^IEN, if non-null all other params are null  ; IBCNESPC("RFLAG")=Report Flag used to indicate which report is being  ; run. Response Report (0), Inactive Report (1), or Ambiguous   ; Report (2).  ; IBCNESPC("DTEXP")=Expiration date used in the inactive policy report  ;  ; Only call this routine at a tag  Q EN(IPRF) ; Main entry pt  ; Init vars  N STOP,IBCNERTN,POP,IBCNESPC  S IBCNESPC("RFLAG")=$G(IPRF)  ;  S STOP=0  S IBCNERTN="IBCNERP1"  W @IOF  W !,"eIV ",$S(IPRF=1:"Inactive Policy",IPRF=2:"Ambiguous Policy",1:"Response")," Report",!  I $G(IPRF) D  . W !,"Please select a date range to view ",$S(IPRF=1:"inactive",1:"ambiguous")," policy information that the eIV"  . W !,"process turned up while attempting to discover previously unknown"  . W !,"insurance policies. (Date range selection is based on the date that"  . W !,"eIV receives the response from the payer.)"  ;  I '$G(IPRF) D  . W !,"Insurance verification responses are received daily."  . W !,"Please select a date range in which responses were received to view the"  . W !,"associated response detail. Otherwise, select a Trace # to view specific"  . W !,"response detail."  ;  ; Rpt by Date Range or Trace # R05 I '$G(IPRF) D RTYPE I STOP G:$$STOP EXIT G R05  ; If rpt by Trace # - no other criteria is necessary  I $G(IBCNESPC("TRCN")) G R100  ; Date Range params R10 D DTRANGE I STOP G:$$STOP EXIT G R05  ; Payer Selection param R20 D PYRSEL I STOP G:$$STOP EXIT G R10  ; Patient Selection param R30 D PTSEL I STOP G:$$STOP EXIT G R20  ; Type of data to return param R40 D TYPE I STOP G:$$STOP EXIT G R30  ; How far back do you want the expiration date R45 I $G(IPRF)=1 D DTEXP I STOP G:$$STOP EXIT G R40  ; Sort by param - Payer or Patient R50 D SORT I STOP G:$$STOP EXIT G R45  ; Select output device R100 D DEVICE(IBCNERTN,.IBCNESPC) I STOP G:$$STOP EXIT G:$G(IBCNESPC("TRCN"))'="" R05 G R50  G EXIT  ; EXIT ; Exit pt  Q  ;  ; COMPILE(IBCNERTN,IBCNESPC) ;   ; Entry point called from EN^XUTMDEVQ in either direct or queued mode.  ; Input params:  ; IBCNERTN = Routine name for ^TMP($J,...  ; IBCNESPC = Array passed by ref of the report params  ;  ; Init scratch globals  K ^TMP($J,IBCNERTN),^TMP($J,IBCNERTN\_"X")  ; Compile  I IBCNERTN="IBCNERP1" D EN^IBCNERP2(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERP4" D EN^IBCNERP5(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERP7" D EN^IBCNERP8(IBCNERTN,.IBCNESPC)  I IBCNERTN="IBCNERPF" D EN^IBCNERPG(IBCNERTN,.IBCNESPC)  ; Print  I '$G(ZTSTOP) D  . I IBCNERTN="IBCNERP1" D EN3^IBCNERPA(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERP4" D EN6^IBCNERPA(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERP7" D EN^IBCNERP9(IBCNERTN,.IBCNESPC)  . I IBCNERTN="IBCNERPF" D EN^IBCNERPH(IBCNERTN,.IBCNESPC)  ; Close device  D ^%ZISC  ; Kill scratch globals  K ^TMP($J,IBCNERTN),^TMP($J,IBCNERTN\_"X")  ; Purge task record  I $D(ZTQUEUED) S ZTREQ="@"  ; COMPILX ; COMPILE exit pt  Q  ; STOP() ; Determine if user wants to exit out of the whole option  ; Init vars  N DIR,X,Y,DIRUT  ;  W !  S DIR(0)="Y"  S DIR("A")="Do you want to exit out of this option entirely"  S DIR("B")="YES"  S DIR("?",1)=" Enter YES to immediately exit out of this option."  S DIR("?")=" Enter NO to return to the previous question."  D ^DIR K DIR  I $D(DIRUT) S (STOP,Y)=1 G STOPX  I 'Y S STOP=0  ; STOPX ; STOP exit pt  Q Y  ; DTRANGE ; Determine start and end dates for date range param  ; Init vars  N X,Y,DIRUT  ;  W !  ;  S DIR(0)="D^:-NOW:EX"  S DIR("A")="Start DATE"  S DIR("?",1)=" Please enter a valid date for which an eIV Response"  S DIR("?")=" would have been received. Future dates are not allowed."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTRANGX  S IBCNESPC("BEGDT")=Y  ; End date DTRANG1 S DIR(0)="DA^"\_Y\_":-NOW:EX"  S DIR("A")=" End DATE: "  S DIR("?",1)=" Please enter a valid date for which an eIV Response"  S DIR("?",2)=" would have been received. This date must not precede"  S DIR("?")=" the Start Date. Future dates are not allowed."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTRANGX  S IBCNESPC("ENDDT")=Y  ; DTRANGX ; DTRANGE exit pt  Q  ; PYRSEL ; Select one payer or ALL - File #365.12  ; Init vars  NEW DIC,DTOUT,DUOUT,X,Y  ;  W !  S DIC(0)="ABEQ"  S DIC("A")=$$FO^IBCNEUT1("Payer or <Return> for All Payers: ",40,"R")  ; Do not allow selection of '~NO PAYER' and non-eIV payers  S DIC("S")="I ($P(^(0),U,1)'=""~NO PAYER""),$$PYRAPP^IBCNEUT5(""IIV"",$G(Y))'="""""  S DIC="^IBE(365.12,"  D ^DIC  I $D(DUOUT)!$D(DTOUT) S STOP=1 G PYRSELX  ; If nothing was selected (Y=-1), select ALL payers  S IBCNESPC("PYR")=$S(Y=-1:"",1:$P(Y,U,1))  ; PYRSELX ; PYRSEL exit pt  Q  ; PTSEL ; Select one patient or ALL - File #2  ; Init vars  NEW DIC,DTOUT,DUOUT,X,Y  ; Patient lookup  W !  S DIC(0)="AEQM"  S DIC("A")=$$FO^IBCNEUT1("Patient or <Return> for All Patients: ",40,"R")  S DIC="^DPT("  D ^DIC  I $D(DUOUT)!$D(DTOUT) S STOP=1 G PTSELX  ; If nothing was selected (Y=-1), select ALL patients  S IBCNESPC("PAT")=$S(Y=-1:"",1:$P(Y,U,1))  ; PTSELX ; PTSEL exit pt  Q  ; TYPE ; Prompt to select to display All or Most Recent Responses for  ; Patient/Payer combos  ; Init vars  N DIR,X,Y,DIRUT  ;  S DIR(0)="S^A:All Responses;M:Most Recent Responses"  S DIR("A")="Select the type of responses to display"  S DIR("B")="A"  S DIR("?",1)=" A - All responses from the payer during the date range will be"  S DIR("?",2)=" displayed for each unique payer/patient combination."  S DIR("?",3)=" (Default)"  S DIR("?",4)=" M - Only the most recently received response from the payer"  S DIR("?",5)=" during the date range will be displayed for each unique"  S DIR("?")=" payer/patient combination."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G TYPEX  S IBCNESPC("TYPE")=Y  ; TYPEX ; TYPE exit pt  Q  ; DTEXP ; Prompt for oldest expiration date to pull for.  ; Init Vars  N Y,DIRUT,TODAY  ;  W !  ;  S DIR(0)="D^:-NOW:EX"  S DIR("A")="Earliest Policy Expiration Date to Select From"  S DIR("B")="T-365"  S DIR("?",1)=" Please enter a valid date in the past. Any policy with a reported"  S DIR("?")=" expiration date prior to this date will not be selected."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G DTEXPX  S IBCNESPC("DTEXP")=Y  ; DTEXPX ; DTEXP Exit  Q  ; SORT ; Prompt to allow users to sort the report by Payer(default) or   ; Patient  ; Init vars  N DIR,X,Y,DIRUT  ;  S DIR(0)="S^1:Payer Name;2:Patient Name"  S DIR("A")="Select the primary sort field"  S DIR("B")=1  S DIR("?",1)=" 1 - Payer Name is the primary sort, Patient Name is secondary."  S DIR("?",2)=" (Default)"  S DIR("?")=" 2 - Patient Name is the primary sort, Payer Name is secondary."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G SORTX  S IBCNESPC("SORT")=Y  ; SORTX ; SORT exit pt  Q  ; RTYPE ; Prompt to allow users to report by date range or Trace #  ; Init vars  N D,DIC,DIR,X,Y,DIRUT,DTOUT,DUOUT  ;  S DIR(0)="S^1:Report by Date Range;2:Report by Trace #"  S DIR("A")="Select the type of report to generate"  S DIR("B")=1  S DIR("?",1)=" 1 - Generate report by date range, payer range, patient range"  S DIR("?",2)=" and All or Most Recent responses for payer/patient."  S DIR("?",3)=" (Default)"  S DIR("?",4)=" 2 - Generate report for a specific Trace # which corresponds"  S DIR("?")=" to an unique response."  D ^DIR K DIR  I $D(DIRUT) S STOP=1 G RTYPEX  I Y=1 S IBCNESPC("TRCN")="" G RTYPEX  ;  ; Allow user to select Trace # from x-ref "C"  W !  S DIC(0)="AEVZSQ"  S DIC="^IBCN(365,",D="C",DIC("A")="Enter Trace # for report: "  S DIC("W")="N IBX S IBX=$P($G(^(0)),U,2,3) W:$P(IBX,U,1) $P($G(^DPT($P(IBX,U,1),0)),U,1) W:$P(IBX,U,2) "" ""\_$P($G(^IBE(365.12,$P(IBX,U,2),0)),U,1)"  D IX^DIC K DIC  I $D(DTOUT)!$D(DUOUT) S STOP=1 G RTYPEX  I 'Y!(Y<0) S STOP=1 G RTYPEX  S IBCNESPC("TRCN")=$P(Y(0),U,9)\_"^"\_$P(Y,U,1)  ; RTYPEX ; RTYPE exit pt  Q  ; DEVICE(IBCNERTN,IBCNESPC) ; Device Handler and possible TaskManager calls  ;  ; Input params:  ; IBCNERTN = Routine name for ^TMP($J,...  ; IBCNESPC = Array passed by ref of the report params  ;  ; Init vars  N ZTRTN,ZTDESC,ZTSAVE,POP  ;  I IBCNERTN="IBCNERP4"!(IBCNERTN="IBCNERPF"&($G(IBCNESPC("TYPE"))="D")) W !!!,"\*\*\* This report is 132 characters wide \*\*\*",!  S ZTRTN="COMPILE^IBCNERP1("""\_IBCNERTN\_""",.IBCNESPC)"  **; RSD 2.6.15.1 – Change Insurance Update to Auto Update**   S ZTDESC="IBCNE eIV "\_$S(IBCNERTN="IBCNERP1":"Response",IBCNERTN="IBCNERPF":"Insurance Update",1:"Payer")\_" Report"  S ZTSAVE("IBCNESPC(")=""  S ZTSAVE("IBCNERTN")=""  **; RSD 2.6.15.14 – Allow user ability to export data to delimited file usable by Microsoft Excel**  **; Code to implement this ability**   D EN^XUTMDEVQ(ZTRTN,ZTDESC,.ZTSAVE)  I POP S STOP=1  ; DEVICEX ; DEVICE exit pt  Q  ; |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNERPG | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.15.3, 2.6.15.7, 2.6.15.8, 2.6.15.9, 2.6.15.10, 2.6.15.11, 2.6.15.12 | | | | | | | | |
| **Related Options** | eIV Patient Insurance Update Report [IBCNE EIV UPDATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNERP1 | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNERPG ;BP/YMG - IBCNE EIV INSURANCE UPDATE REPORT COMPILE;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; variables from IBCNERPF:  ; IBCNERTN = "IBCNERPF"  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  ; Output variables passed to IBCNERPH:  ; Summary report:  ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Count  ; SORT1 - Payer Name or \*, SORT2 - Clerk Name or 0 if not processed  ;  ; Detailed report:  ; ^TMP($J,IBCNERTN,SORT1)=Count   ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Pat. Name ^ SSN ^ Date received ^ Payer Name ^ Ck AB ^ Clerk Name ^ Date Verified ^ Days old  ; SORT1 - Payer Name, Patient Name, or Clerk Name, SORT2 - Date received  ;  Q  ; EN(IBCNERTN,IBCNESPC) ; Entry point  N ALLPYR,ALLPAT,DATE,BDATE,EDATE,RPDATA,RTYPE,SORT  S ALLPYR=$S($G(IBCNESPC("PYR"))="A":1,1:0)  S ALLPAT=$S($G(IBCNESPC("PAT"))="A":1,1:0)  S BDATE=$G(IBCNESPC("BEGDT"))  S EDATE=$G(IBCNESPC("ENDDT"))  I EDATE'="",$P(EDATE,".",2)="" S EDATE=$$FMADD^XLFDT(EDATE,0,23,59,59)  S RTYPE=$G(IBCNESPC("TYPE"))  S SORT=$G(IBCNESPC("SORT"))  I '$D(ZTQUEUED),$G(IOST)["C-" W !!,"Compiling report data ..."  ; Kill scratch global  K ^TMP($J,IBCNERTN)  S DATE=$O(^IBCN(365,"AD",BDATE),-1)  F  S DATE=$O(^IBCN(365,"AD",DATE)) Q:'DATE!(DATE>EDATE) D PAYERS(DATE,ALLPYR,ALLPAT) Q:$G(ZTSTOP)  M ^TMP($J,IBCNERTN)=RPDATA  Q  ; PAYERS(DATE,ALLPYR,ALLPAT) ; loop through payers  N PYR  S PYR=""  I 'ALLPYR F  S PYR=$O(IBCNESPC("PYR",PYR)) Q:'PYR  D:$O(^IBCN(365,"AD",DATE,PYR,"")) PATIENTS(DATE,PYR,ALLPAT) Q:$G(ZTSTOP)  I ALLPYR F  S PYR=$O(^IBCN(365,"AD",DATE,PYR)) Q:'PYR  D PATIENTS(DATE,PYR,ALLPAT) Q:$G(ZTSTOP)  Q  ; PATIENTS(DATE,PYR,ALLPAT) ; loop through patients  N PAT  S PAT=""  I 'ALLPAT F  S PAT=$O(IBCNESPC("PAT",PAT)) Q:'PAT  D:$O(^IBCN(365,"AD",DATE,PYR,PAT,"")) GETDATA(DATE,PYR,PAT) Q:$G(ZTSTOP)  I ALLPAT F  S PAT=$O(^IBCN(365,"AD",DATE,PYR,PAT)) Q:'PAT  D GETDATA(DATE,PYR,PAT) Q:$G(ZTSTOP)  Q  ; GETDATA(DATE,PYR,PAT) ; loop through responses and compile report  N ABDATE,ABIEN,AUTOUPD,CHKAB,CLNAME,GIEN,IENS2,IENS312,INS,NOW,PATNAME,PYRNAME,RIEN,SORT1,SORT2,SSN,TOTMES,TQ,VDATE  ;  S NOW=$$NOW^XLFDT  S (TOTMES,INS)=0  S RIEN="" F  S RIEN=$O(^IBCN(365,"AD",DATE,PYR,PAT,RIEN)) Q:'RIEN  D  Q:$G(ZTSTOP)  .S TOTMES=TOTMES+1 I $D(ZTQUEUED),TOTMES#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  .S TQ=+$P(^IBCN(365,RIEN,0),U,5) I TQ S INS=+$P(^IBCN(365.1,TQ,0),U,13)  .I 'INS Q  .S IENS2=PAT\_",",IENS312=INS\_","\_IENS2  .S VDATE=$$GET1^DIQ(2.312,IENS312,1.03,"I") I VDATE=""!(VDATE<BDATE)!(VDATE>EDATE) Q  .S PYRNAME=$P(^IBE(365.12,PYR,0),U),PATNAME=$$GET1^DIQ(2,IENS2,.01,"E")  .S AUTOUPD=+$$GET1^DIQ(2.312,IENS312,4.04,"I")  .I AUTOUPD S CLNAME="AUTOUPDATE,IB-eIV"  .I 'AUTOUPD S CLNAME=$$GET1^DIQ(2.312,IENS312,1.04,"E") I CLNAME="" S CLNAME="UNKNOWN"  .I RTYPE="S" S SORT1=$S(ALLPYR:"\*",1:PYRNAME),SORT2=CLNAME,RPDATA(SORT1,SORT2)=$G(RPDATA(SORT1,SORT2))+1 Q  .S SSN=$$GET1^DIQ(2,IENS2,.09,"E")  .S CHKAB="Y"  .S GIEN=$$GET1^DIQ(2.312,IENS312,.18,"I"),ABIEN="",ABDATE=""  .I +GIEN,$D(^IBA(355.4,"APY",GIEN)) S ABIEN=$O(^IBA(355.4,"APY",GIEN,$O(^IBA(355.4,"APY",GIEN,"")),""))  .S:+ABIEN ABDATE=$P($G(^IBA(355.4,ABIEN,1)),U,3)  .S:+ABDATE CHKAB=$S($$FMDIFF^XLFDT(NOW,ABDATE)>365:"Y",1:"N")  .S SORT1=$S(SORT=1:PYRNAME,SORT=2:PATNAME,1:CLNAME),SORT2=DATE  .S RPDATA(SORT1)=$G(RPDATA(SORT1))+1   .S RPDATA(SORT1,SORT2)=PATNAME\_U\_SSN\_U\_DATE\_U\_PYRNAME\_U\_CHKAB\_U\_CLNAME\_U\_VDATE\_U\_$$FMDIFF^XLFDT(NOW,DATE) .Q  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNERPG ;BP/YMG - IBCNE EIV INSURANCE UPDATE REPORT COMPILE;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; variables from IBCNERPF:  ; IBCNERTN = "IBCNERPF"  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  ; Output variables passed to IBCNERPH:  ; Summary report:  ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Count  ; SORT1 - Payer Name or \*, SORT2 - Clerk Name or 0 if not processed  ;  ; Detailed report:  ; ^TMP($J,IBCNERTN,SORT1)=Count   ; ^TMP($J,IBCNERTN,SORT1,SORT2)=Pat. Name ^ SSN ^ Date received ^ Payer Name ^ Ck AB ^ Clerk Name ^ Date Verified ^ Days old  ; SORT1 - Payer Name, Patient Name, or Clerk Name, SORT2 - Date received  ;  Q  ; EN(IBCNERTN,IBCNESPC) ; Entry point  N ALLPYR,ALLPAT,DATE,BDATE,EDATE,RPDATA,RTYPE,SORT  S ALLPYR=$S($G(IBCNESPC("PYR"))="A":1,1:0)  S ALLPAT=$S($G(IBCNESPC("PAT"))="A":1,1:0)  S BDATE=$G(IBCNESPC("BEGDT"))  S EDATE=$G(IBCNESPC("ENDDT"))  I EDATE'="",$P(EDATE,".",2)="" S EDATE=$$FMADD^XLFDT(EDATE,0,23,59,59)  S RTYPE=$G(IBCNESPC("TYPE"))  S SORT=$G(IBCNESPC("SORT"))  I '$D(ZTQUEUED),$G(IOST)["C-" W !!,"Compiling report data ..."  ; Kill scratch global  K ^TMP($J,IBCNERTN)  S DATE=$O(^IBCN(365,"AD",BDATE),-1)  F  S DATE=$O(^IBCN(365,"AD",DATE)) Q:'DATE!(DATE>EDATE) D PAYERS(DATE,ALLPYR,ALLPAT) Q:$G(ZTSTOP)  M ^TMP($J,IBCNERTN)=RPDATA  Q  ; PAYERS(DATE,ALLPYR,ALLPAT) ; loop through payers N PYR,**UPDTFLG,USRSELIC . . ;2.6.15.3, 2.6.15.8 Added variables  S PYR=""  I 'ALLPYR D  Q  . F  D  Q:'PYR  Q:$G(ZTSTOP)  . . S PYR=$O(IBCNESPC("PYR",PYR))  . . Q:'PYR  . . ;2.6.15.3 S UPDTFLG=Payer's Auto Update Flag  . . Q:'UPDTFLG  . . ;2.6.15.8 S USRSELIC=User Selected Insurance Flag  . . Q:'USRSELIC  . . D:$O(^IBCN(365,"AD",DATE,PYR,"")) PATIENTS(DATE,PYR,ALLPAT)  ;  ; All Payers selected  F  D  Q:'PYR  Q:$G(ZTSTOP)  . S PYR=$O(^IBCN(365,"AD",DATE,PYR))  . Q:'PYR  . ;2.6.15.3 S UPDTFLG=Payer's Auto Update Flag  . Q:'UPDTFLG  . ;2.6.15.8 S USRSELIC=User Selected Insurance Flag  . Q:'USRSELIC  . D PATIENTS(DATE,PYR,ALLPAT)**   Q  ; PATIENTS(DATE,PYR,ALLPAT) ; loop through patients  N PAT  S PAT=""  I 'ALLPAT F  S PAT=$O(IBCNESPC("PAT",PAT)) Q:'PAT  D:$O(^IBCN(365,"AD",DATE,PYR,PAT,"")) GETDATA(DATE,PYR,PAT) Q:$G(ZTSTOP)  I ALLPAT F  S PAT=$O(^IBCN(365,"AD",DATE,PYR,PAT)) Q:'PAT  D GETDATA(DATE,PYR,PAT) Q:$G(ZTSTOP)  Q  ; GETDATA(DATE,PYR,PAT) ; loop through responses and compile report **; RSD 2.6.15.8 – Only include user-selected insurance companies associated with payer (with total count of auto-updated policies for that company)**  **; RSD 2.6.15.9 - Summary version of the report shall include a total count of auto updated policies per selected payer;**  **; however, if insurance companies were selected by the user for that payer then the count for that payer is the sum of the auto**  **; updated policies for only the selected associated insurance companies.**  **; RSD 2.6.15.10 – Report is sorted by the Payer’s name**   N ABDATE,ABIEN,AUTOUPD,CHKAB,CLNAME,GIEN,IENS2,IENS312,INS,NOW,PATNAME,PYRNAME,RIEN,SORT1,SORT2,SSN,TOTMES,TQ,VDATE  ;  S NOW=$$NOW^XLFDT  S (TOTMES,INS)=0  S RIEN="" F  S RIEN=$O(^IBCN(365,"AD",DATE,PYR,PAT,RIEN)) Q:'RIEN  D  Q:$G(ZTSTOP)  .S TOTMES=TOTMES+1 I $D(ZTQUEUED),TOTMES#100=0,$$S^%ZTLOAD() S ZTSTOP=1 Q  .S TQ=+$P(^IBCN(365,RIEN,0),U,5) I TQ S INS=+$P(^IBCN(365.1,TQ,0),U,13)  .I 'INS Q  .S IENS2=PAT\_",",IENS312=INS\_","\_IENS2  .S VDATE=$$GET1^DIQ(2.312,IENS312,1.03,"I") I VDATE=""!(VDATE<BDATE)!(VDATE>EDATE) Q  .S PYRNAME=$P(^IBE(365.12,PYR,0),U),PATNAME=$$GET1^DIQ(2,IENS2,.01,"E")  .S AUTOUPD=+$$GET1^DIQ(2.312,IENS312,4.04,"I") ~~.I AUTOUPD S CLNAME="AUTOUPDATE,IB-eIV"~~ **;2.6.15.7 remove line**  **. Q:AUTOUPD ;2.6.15.7 add line**  .**~~I 'AUTOUPD~~**S CLNAME=$$GET1^DIQ(2.312,IENS312,1.04,"E") I CLNAME="" S CLNAME="UNKNOWN"  .I RTYPE="S" S SORT1=$S(ALLPYR:"\*",1:PYRNAME),SORT2=CLNAME,RPDATA(SORT1,SORT2)=$G(RPDATA(SORT1,SORT2))+1 Q  .S SSN=$$GET1^DIQ(2,IENS2,.09,"E")  .S CHKAB="Y"  .S GIEN=$$GET1^DIQ(2.312,IENS312,.18,"I"),ABIEN="",ABDATE=""  .I +GIEN,$D(^IBA(355.4,"APY",GIEN)) S ABIEN=$O(^IBA(355.4,"APY",GIEN,$O(^IBA(355.4,"APY",GIEN,"")),""))  .S:+ABIEN ABDATE=$P($G(^IBA(355.4,ABIEN,1)),U,3)  .S:+ABDATE CHKAB=$S($$FMDIFF^XLFDT(NOW,ABDATE)>365:"Y",1:"N")  .S SORT1=$S(SORT=1:PYRNAME,SORT=2:PATNAME,1:CLNAME),SORT2=DATE  .S RPDATA(SORT1)=$G(RPDATA(SORT1))+1  **. ; 2.6.15.11 Remove DATE, CHKAB, VDATE**  **. ; 2.6.15.12 Add Insurance Company Name, Date EIV Inquiry sent, Auto policy  . ; update date and trace number**   .S RPDATA(SORT1,SORT2)=PATNAME\_U\_SSN\_**~~U\_DATE~~**\_U\_PYRNAME\_**~~U\_CHKAB~~**\_U\_CLNAME\_**~~U\_VDATE~~**\_U\_$$FMDIFF^XLFDT(NOW,DATE) .Q  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | IBCNEPRF | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.15.1, 2.6.15.2, 2.6.15.4, 2.6.15.5, 2.6.15.6 | | | | | | | | |
| **Related Options** | eIV Patient Insurance Update Report [IBCNE EIV UPDATE REPORT] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | None | | | | | STOP^IBCNERP1  DEVICE^IBCNERP1 | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNERPF ;BP/YMG - IBCNE USER INTERFACE EIV INSURANCE UPDATE REPORT ;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; Variables:  ; IBCNERTN = "IBCNERPF" (current routine name for queueing the   ; COMPILE process)  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  Q EN ; entry point  N STOP,IBCNERTN,IBCNESPC  ;  S STOP=0,IBCNERTN="IBCNERPF"  W @IOF  W !,"eIV Insurance Update Report",!  ; Prompts for Insurance Update Report  ; Report Type - Summary or Detailed P10 D TYPE I STOP G EXIT  ; Payer Selection parameter P20 D PAYER I STOP G:$$STOP^IBCNERP1 EXIT G P10  ; Date Range parameters P30 D DTRANGE I STOP G:$$STOP^IBCNERP1 EXIT G P20  ; Patient Selection parameter P40 D PATIENT I STOP G:$$STOP^IBCNERP1 EXIT G P30  ; Sort by parameter - Payer Name, Patient Name, or Clerk Name P50 D SORT I STOP G:$$STOP^IBCNERP1 EXIT G P40  ; Select the output device P100 D DEVICE^IBCNERP1(IBCNERTN,.IBCNESPC) I STOP G:$$STOP^IBCNERP1 EXIT G P50  ; EXIT ;  Q  ; PAYER ;  N DIC,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR("A")="Run for (A)ll Payers or (S)elected Payers: "  S DIR("A",1)="PAYER SELECTION:"  S DIR(0)="SA^A:All;S:Selected",DIR("B")="A"  D ^DIR  I $D(DIRUT) S STOP=1 Q  I Y="A" S IBCNESPC("PYR")="A" Q  ; "All Payers" selected  S DIC(0)="ABEQ"  S DIC("A")="Select Insurance Company: "  ; Do not allow selection of '~NO PAYER' and non-eIV payers  S DIC("S")="I ($P(^(0),U,1)'=""~NO PAYER""),$$PYRAPP^IBCNEUT5(""IIV"",$G(Y))'="""""  S DIC="^IBE(365.12," PAYER1 ;  D ^DIC  I $D(DUOUT)!$D(DTOUT)!(Y=-1) S STOP=1 K IBCNESPC("PYR") Q  S IBCNESPC("PYR",$P(Y,U,1))=""  I $$ANOTHER G PAYER1  Q  ; DTRANGE ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR(0)="D^::EX",DIR("B")="Today"  S DIR("A")="Earliest Date Received"  S DIR("A",1)="RESPONSE RECEIVED DATE RANGE SELECTION:"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("BEGDT")=Y  ; End date DTRANGE1 ;  K DIR("A") S DIR("A")=" Latest Date Received"  D ^DIR I $D(DIRUT) S STOP=1 Q  I Y<IBCNESPC("BEGDT") W !," Latest Date must not precede the Earliest Date." G DTRANGE1  S IBCNESPC("ENDDT")=Y  Q  ; PATIENT ;  N DIC,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  ; summary report is always run for all patients  I $G(IBCNESPC("TYPE"))="S" S IBCNESPC("PAT")="A" Q  W !  S DIR("A")="Run for (A)ll Patients or (S)elected Patients: "  S DIR("A",1)="PATIENT SELECTION:"  S DIR(0)="SA^A:All;S:Selected",DIR("B")="A"  D ^DIR  I $D(DIRUT) S STOP=1 Q  I Y="A" S IBCNESPC("PAT")="A" Q  ; "All Patients" selected  S DIC(0)="ABEQ"  S DIC("A")="Select Patient: "  S DIC="^DPT(" PATIENT1 ;  D ^DIC  I $D(DUOUT)!$D(DTOUT)!(Y=-1) S STOP=1 K IBCNESPC("PAT") Q  S IBCNESPC("PAT",$P(Y,U,1))=""  I $$ANOTHER G PATIENT1  Q  ; ANOTHER() ; "Select Another" prompt  ; returns 1, if response was "YES", returns 0 otherwise  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DIR("A")="Select Another?" S DIR(0)="Y",DIR("B")="NO"  D ^DIR I $D(DIRUT) S STOP=1  Q Y  ; SORT ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  ; summary report is sorted by Payer Name, if run for selected payers,  ; or by Clerk Name, if run for all payers  I $G(IBCNESPC("TYPE"))="S" S IBCNESPC("SORT")=$S($G(IBCNESPC("PYR"))="A":1,1:3) Q  W !  S DIR(0)="SA^1:Payer Name;2:Patient Name;3:Clerk Name"  S DIR("A")="Sort By: "  S DIR("A",1)="SORT CRITERIA:"  S DIR("B")="Payer Name"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("SORT")=Y  Q  ; TYPE ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR(0)="SA^S:Summary;D:Detailed"  S DIR("A")="Run a (S)ummary or (D)etailed Report: "  S DIR("B")="Summary"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("TYPE")=Y  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNERPF ;BP/YMG - IBCNE USER INTERFACE EIV INSURANCE UPDATE REPORT ;16-SEP-2009  ;;2.0;INTEGRATED BILLING;\*\*416\*\*;16-SEP-09;Build 58  ;;Per VHA Directive 2004-038, this routine should not be modified.  ;  ; Variables:  ; IBCNERTN = "IBCNERPF" (current routine name for queueing the   ; COMPILE process)  ; IBCNESPC("BEGDT") = start date for date range  ; IBCNESPC("ENDDT") = end date for date range  ; IBCNESPC("PYR",ien) = payer iens for report, if IBCNESPC("PYR")="A", then include all  ; IBCNESPC("PAT",ien) = patient iens for report, if IBCNESPC("PAT")="A", then include all  ; IBCNESPC("SORT") = sort by: 1 - Payer name, 2 - Patient Name, 3 - Clerk Name  ; IBCNESPC("TYPE") = report type: "S" - summary, "D" - detailed  ;  Q EN ; entry point  N STOP,IBCNERTN,IBCNESPC  ;  S STOP=0,IBCNERTN="IBCNERPF"  W @IOF **; RSD 2.6.15.1 – Rename eIV Patient Insurance Update Report to eIV Auto Update Report**  W !,"eIV **~~Insurance~~** **Auto** Update Report",!  ; Prompts for **~~Insurance~~** **Auto** Update Report  ; Report Type - Summary or Detailed P10 D TYPE I STOP G EXIT  ; Payer Selection parameter P20 D PAYER I STOP G:$$STOP^IBCNERP1 EXIT G P10  ; Date Range parameters P30 D DTRANGE I STOP G:$$STOP^IBCNERP1 EXIT G P20  ; Patient Selection parameter P40 D PATIENT I STOP G:$$STOP^IBCNERP1 EXIT G P30  ; Sort by parameter - Payer Name, Patient Name, or Clerk Name P50 D SORT I STOP G:$$STOP^IBCNERP1 EXIT G P40  ; Select the output device P100 D DEVICE^IBCNERP1(IBCNERTN,.IBCNESPC) I STOP G:$$STOP^IBCNERP1 EXIT G P50  ; EXIT ;  Q  ; PAYER ;  N DIC,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR("A")="Run for (A)ll Payers or (S)elected Payers: "  S DIR("A",1)="PAYER SELECTION:"  S DIR(0)="SA^A:All;S:Selected",DIR("B")="A"  D ^DIR  I $D(DIRUT) S STOP=1 Q  I Y="A" S IBCNESPC("PYR")="A" Q  ; "All Payers" selected  S DIC(0)="ABEQ" **; RSD 2.6.15.2 – Change prompt from ‘Select Insurance Company’ to ‘Select Payer’**   S DIC("A")="Select **~~Insurance Company~~Payer**: "  ; Do not allow selection of '~NO PAYER' and non-eIV payers  S DIC("S")="I ($P(^(0),U,1)'=""~NO PAYER""),$$PYRAPP^IBCNEUT5(""IIV"",$G(Y))'="""""  S DIC="^IBE(365.12," PAYER1 ; **; RSD 2.6.15.4– Allow the user the choice of whether to display associated insurance companies or not**  **; RSD 2.6.15.5 – If user chooses to display insurance companies, system shall prompt user for companies filtering only those linked to payer**  **; RSD 2.6.15.6 – System shall allow user to select 0, 1 or multiple companies associated with given payer**   D ^DIC  I $D(DUOUT)!$D(DTOUT)!(Y=-1) S STOP=1 K IBCNESPC("PYR") Q  S IBCNESPC("PYR",$P(Y,U,1))=""  I $$ANOTHER G PAYER1  Q  ; DTRANGE ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR(0)="D^::EX",DIR("B")="Today"  S DIR("A")="Earliest Date Received"  S DIR("A",1)="RESPONSE RECEIVED DATE RANGE SELECTION:"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("BEGDT")=Y  ; End date DTRANGE1 ;  K DIR("A") S DIR("A")=" Latest Date Received"  D ^DIR I $D(DIRUT) S STOP=1 Q  I Y<IBCNESPC("BEGDT") W !," Latest Date must not precede the Earliest Date." G DTRANGE1  S IBCNESPC("ENDDT")=Y  Q  ; PATIENT ;  N DIC,DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  ; summary report is always run for all patients  I $G(IBCNESPC("TYPE"))="S" S IBCNESPC("PAT")="A" Q  W !  S DIR("A")="Run for (A)ll Patients or (S)elected Patients: "  S DIR("A",1)="PATIENT SELECTION:"  S DIR(0)="SA^A:All;S:Selected",DIR("B")="A"  D ^DIR  I $D(DIRUT) S STOP=1 Q  I Y="A" S IBCNESPC("PAT")="A" Q  ; "All Patients" selected  S DIC(0)="ABEQ"  S DIC("A")="Select Patient: "  S DIC="^DPT(" PATIENT1 ;  D ^DIC  I $D(DUOUT)!$D(DTOUT)!(Y=-1) S STOP=1 K IBCNESPC("PAT") Q  S IBCNESPC("PAT",$P(Y,U,1))=""  I $$ANOTHER G PATIENT1  Q  ; ANOTHER() ; "Select Another" prompt  ; returns 1, if response was "YES", returns 0 otherwise  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  S DIR("A")="Select Another?" S DIR(0)="Y",DIR("B")="NO"  D ^DIR I $D(DIRUT) S STOP=1  Q Y  ; SORT ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  ; summary report is sorted by Payer Name, if run for selected payers,  ; or by Clerk Name, if run for all payers  I $G(IBCNESPC("TYPE"))="S" S IBCNESPC("SORT")=$S($G(IBCNESPC("PYR"))="A":1,1:3) Q  W !  S DIR(0)="SA^1:Payer Name;2:Patient Name;3:Clerk Name"  S DIR("A")="Sort By: "  S DIR("A",1)="SORT CRITERIA:"  S DIR("B")="Payer Name"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("SORT")=Y  Q  ; TYPE ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,X,Y  W !  S DIR(0)="SA^S:Summary;D:Detailed"  S DIR("A")="Run a (S)ummary or (D)etailed Report: "  S DIR("B")="Summary"  D ^DIR I $D(DIRUT) S STOP=1 Q  S IBCNESPC("TYPE")=Y  Q |

##### Options

| Options | Instructions |
| --- | --- |
| **Option Name (MENU TEXT field)** | Missing Data Report |
| **Option Name (NAME field)** | IBCN MISSING DATA |
| **Enhancement Category** | New  Modify  Delete  No Change |
| **RTM** | 2.6.12.1 |
| **Data Passing** | N/A |
| **Menu Text Description** | This option will list all selected insurance companies that contain one or more of the selected missing data elements. |
| **Option Type** | run routine |
| **Option Definition** | **Routine:** IBCMDT  **Package:** Integrated Billing |
| **Current Entry Action Logic** | N/A |
| **Modified Entry Action Logic (Changes are in bold)** | N/A |
| **Current Exit Action Logic** | N/A |
| **Modified Exit Action Logic (Changes are in bold)** | N/A |

| Options | Instructions |
| --- | --- |
| **Option Name (MENU TEXT field)** | eIV Patient Insurance Update Report |
| **Option Name (NAME field)** | IBCNE EIV UPDATE REPORT |
| **Enhancement Category** | New  Modify  Delete  No Change |
| **RTM** | 2.6.15.1 |
| **Data Passing** | N/A |
| **Menu Text Description** | Generate eIV Patient Insurance Update Report based on eIV Inquiries and Responses for a given date range and current Patient Insurance. |
| **Option Type** | run routine |
| **Option Definition** | **Routine:** EN^IBCNERPF  **New Option Name:** eIV Auto Update Report  **Package:** Integrated Billing |
| **Current Entry Action Logic** | N/A |
| **Modified Entry Action Logic (Changes are in bold)** | N/A |
| **Current Exit Action Logic** | N/A |
| **Modified Exit Action Logic (Changes are in bold)** | N/A |

#### System Feature: Worklist – Move Subscribers to a Different Plan

RSD 2.6.16.1 Worklist (Move Subscribers to a Different Plan) - The worklist shall include a list of subscribers if the user answers they do NOT want to move the entire group.

RSD 2.6.16.2 Worklist (Move Subscribers to a Different Plan) - The worklist shall prompt a user with the following filters if the user answers they do NOT want to move the entire group:   
- Do you want to filter out deceased patients? (yes/no);   
- Do you want to filter subscriber ID? (yes/no);  
- Filter subscriber IDs that contain: ;   
- Do you want to filter for active or inactive policies? (yes/no);   
- List of active or inactive policies;   
- Do you want to filter effective dates? (yes/no);  
- Prompt for Effective date range

RSD 2.6.16.3 Worklist (Move Subscribers to a Different Plan) - The ListMan lookup for subscribers shall display the following data elements:   
- Header contains "Patient Policy Lookup"  
- Count of subscribers that met filter requirements  
- Patient Name, last 4 SSN, DOB, Subscriber ID, Effective Date, Expiration Date, “Whose” policy, (and Patient ID if room on screen – keep at 1 line of data)

RSD 2.6.16.4 Worklist (Move Subscribers to a Different Plan) - The ListMan lookup for subscribers shall allow a user to run a Search.

RSD 2.6.16.5 Worklist (Move Subscribers to a Different Plan) - The header for the group plan ListMan lookup shall be changed from "Insurance Plan Lookup" to "Group Plan Lookup".

RSD 2.6.16.6 Worklist (Move Subscribers to a Different Plan) - The report shall allow a user to select from the list of filtered subscribers, one or more subscribers to be moved, including a range (i.e. 1, 4-5, 7).

#### Design Element Tables (Worklist- Move Subscribers to a Different Plan)

##### Routines (Entry Points)

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSUR** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.16.5 | | | | | | | | |
| **Related Options** | Move Subscribers to a Different Plan [IBCN MOVE SUBSCRIB TO PLAN] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNSUR1 | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .  SEL(IBNP) ; Select a company and plan.  ; Input: IBNP -- If set to 1, allows adding a new plan and  ; -- Screen Inactive Companies  ; -- If set to 0, must have at least one group plan  ; Output: IBCNS -- Pointer to selected company in file #36  ; IBPLAN -- Pointer to selected/added plan in file #355.3  ; IBQUIT -- Set to 1 if the user wants to quit.  ;  N X,Y K DIC,DIR  S DIC(0)="QEAMZ",DIC="^DIC(36,"  I 'IBNP S DIC("S")="I $$ANYGP^IBCNSJ(+Y,0,1)"  I IBNP S DIC("S")="I '$P($G(^DIC(36,+Y,0)),U,5)"  S DIC("A")="Select INSURANCE COMPANY: "  D ^DIC K DIC S IBCNS=+Y  I Y<0 W " <No Insurance Company selected>" S IBQUIT=1 G SELQ  ;  ; - if a new plan may be added, allow adding  I IBNP D  I (IBPLAN)!(IBQUIT) G SELQ  .W !!,"You may add a new Plan at this time or select an existing Plan."  .; IB\*2.0\*506 added IBKEY parameter (4th) to the NEW^IBCNSJ3 call (check user's security keys)  .D NEW^IBCNSJ3(IBCNS,.IBPLAN,1,1)  .I 'IBPLAN,'$$ANYGP^IBCNSJ(+IBCNS,0,1) W !!,\*7,"Insurance Company receiving subscribers must have a Plan." S IBQUIT=1  ;  ; - see if user wants to select the plan  W !!,"You may select an existing Plan from a list or enter a specific Plan.",!  S DIR(0)="Y",DIR("B")="YES",DIR("A")="Do you wish to enter a specific plan"  S DIR("?")="The look-up facility to select a group plan has been enhanced to use the List Manager. Enter 'NO' if you wish to select a plan from this look-up, or 'YES' to directly enter a plan."  D ^DIR K DIR I $D(DIRUT) S IBQUIT=1 G SELQ  ;  ; - invoke the plan look-up  I 'Y D  G SELQ  .W " ..." S IBPLAN=0 D LKP^IBCNSU2(IBCNS,0,0,.IBPLAN,0,1)  .I 'IBPLAN W !!,\*7,"\* No plan selected!",! S IBQUIT=1  ;  ; - allow a FileMan look-up  S DIC("A")="Select a GROUP PLAN: "  S DIC="^IBA(355.3,",DIC(0)="AEQM",DIC("S")="I +^(0)=IBCNS,$P(^(0),U,2)"  S DIC("W")="N IBX S IBX=$G(^(0)) W "" Name: "",$E($S($P(IBX,U,3)]"""":$P(IBX,U,3),1:""<none>"")\_$J("""",20),1,20),"" Number: "",$S($P(IBX,U,4)]"""":$P(IBX,U,4),1:""<none>"")"  D ^DIC K DIC S IBPLAN=+Y  I Y<0 W !!,\*7,"\* No plan selected!",! S IBQUIT=1  ; SELQ K DIRUT,DUOUT,DTOUT,DIROUT  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  .   ;  SEL(IBNP) ; Select a company and plan.  ; Input: IBNP -- If set to 1, allows adding a new plan and  ; -- Screen Inactive Companies  ; -- If set to 0, must have at least one group plan  ; Output: IBCNS -- Pointer to selected company in file #36  ; IBPLAN -- Pointer to selected/added plan in file #355.3  ; IBQUIT -- Set to 1 if the user wants to quit.  ;  N X,Y K DIC,DIR  S DIC(0)="QEAMZ",DIC="^DIC(36,"  I 'IBNP S DIC("S")="I $$ANYGP^IBCNSJ(+Y,0,1)"  I IBNP S DIC("S")="I '$P($G(^DIC(36,+Y,0)),U,5)"  S DIC("A")="Select INSURANCE COMPANY: "  D ^DIC K DIC S IBCNS=+Y  I Y<0 W " <No Insurance Company selected>" S IBQUIT=1 G SELQ  ;  ; - if a new plan may be added, allow adding  I IBNP D  I (IBPLAN)!(IBQUIT) G SELQ  .W !!,"You may add a new Plan at this time or select an existing Plan."  .; IB\*2.0\*506 added IBKEY parameter (4th) to the NEW^IBCNSJ3 call (check user's security keys)  .D NEW^IBCNSJ3(IBCNS,.IBPLAN,1,1)  .I 'IBPLAN,'$$ANYGP^IBCNSJ(+IBCNS,0,1) W !!,\*7,"Insurance Company receiving subscribers must have a Plan." S IBQUIT=1  ;  ; - see if user wants to select the plan  W !!,"You may select an existing Plan from a list or enter a specific Plan.",!  S DIR(0)="Y",DIR("B")="YES",DIR("A")="Do you wish to enter a specific plan"  S DIR("?")="The look-up facility to select a group plan has been enhanced to use the List Manager. Enter 'NO' if you wish to select a plan from this look-up, or 'YES' to directly enter a plan."  D ^DIR K DIR I $D(DIRUT) S IBQUIT=1 G SELQ  ;  ; - invoke the plan look-up  I 'Y D  G SELQ  **.S VALM(“TITLE”)="Group Plan Lookup" 2.6.16.5**   .W " ..." S IBPLAN=0 D LKP^IBCNSU2(IBCNS,0,0,.IBPLAN,0,1)  .I 'IBPLAN W !!,\*7,"\* No plan selected!",! S IBQUIT=1  ;  ; - allow a FileMan look-up  S DIC("A")="Select a GROUP PLAN: "  S DIC="^IBA(355.3,",DIC(0)="AEQM",DIC("S")="I +^(0)=IBCNS,$P(^(0),U,2)"  S DIC("W")="N IBX S IBX=$G(^(0)) W "" Name: "",$E($S($P(IBX,U,3)]"""":$P(IBX,U,3),1:""<none>"")\_$J("""",20),1,20),"" Number: "",$S($P(IBX,U,4)]"""":$P(IBX,U,4),1:""<none>"")"  D ^DIC K DIC S IBPLAN=+Y  I Y<0 W !!,\*7,"\* No plan selected!",! S IBQUIT=1  ; SELQ K DIRUT,DUOUT,DTOUT,DIROUT  Q  ; |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSUR1** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.16.1, 2.6.16.2 | | | | | | | | |
| **Related Options** | Move Subscribers to a Different Plan [IBCN MOVE SUBSCRIB TO PLAN] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNSUR | | | | | EDUP^IBCNSA2  SUBS^IBCNSJ  DBU^IBCNSJ  MERG^IBCNSJ13  SWPL^IBCNSJ13  PROC^IBCNSUR  SEL^IBCNSUR  ADS^IBCNSUR3  BHEAD^IBCNSUR3  DONE^IBCNSUR3  DAT2^IBOUTL | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| IBCNSUR1 ;ALB/CPM/CMS - MOVE SUBSCRIBERS TO DIFFERENT PLAN (CON'T)   ;;2.0;INTEGRATED BILLING;\*\*103,225,276\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  Q  ;  ; PROC ; - Top of processing from IBCNSUR  ; Move subscribers to another company's insurance plan.  N IBCNS,IBPLAN,IBC1,IBC1N,IBC1X,IBC2,IBC2N,IBC2X,IBCPOL  N IBP1,IBP1N,IBP1X,IBP2,IBP2N,IBP2X,IBQ,IBQUIT,IBSUB,DFN,IBCDFN  N IBXXX,IBX,IBDAT,IBCDFN1,IBNP,IBAB,IBI,IBIAB,IBCAB,IBW,IBST  N DIC,DIE,DR,DA,D0,DIR,DIRUT,DIROUT,DTOUT,DUOUT,I,X,Y,DIK,DLAYGO  N IBSPLIT,IBEFFDT,IBEXPDT,REF,IBLN  ;  K ^TMP($J,"IBCNSUR") ; subscribers  K ^TMP($J,"IBCNSUR1") ; e-mail bulletin  S REF=$NA(^TMP($J,"IBCNSUR1")),IBLN=0  ;  S IBQUIT=0  W !!!,"=====================",!,"MOVE SUBSCRIBERS FROM",!,"====================="  W !!,"Select the Insurance Company and Plan to move subscribers FROM.",!  ;  ; - select company/plan for subscribers to be moved  S IBQUIT=0  D SEL^IBCNSUR(0)  I IBQUIT S IBSTOP=1 G PROCQ  ;  ; - collect the plan subscribers  S IBC1=IBCNS,IBP1=IBPLAN  W !!,"Collecting Subscribers ..."  S IBSUB=$$SUBS^IBCNSJ(IBC1,IBP1,0,"^TMP($J,""IBCNSUR"")")  I 'IBSUB W !!,?5,\*7,"\* This plan has no subscribers!" S IBQUIT=1 G PROCQ  W !!,"This plan has ",+IBSUB," subscribers. All subscribers will be moved."  ;   ; - select company/plan to move subscribers  W !!!,"MOVE SUBSCRIBERS TO"  W !!,"Select the Insurance Company and Plan to move subscribers TO.",!  D SEL^IBCNSUR(1)  I IBQUIT G PROCQ  I $P($G(^DIC(36,IBCNS,0)),"^",5) W !!,\*7,"You must move the subscribers to an active insurance company!" G PROCQ  S IBC2=IBCNS,IBP2=IBPLAN  ;  ; - make sure not moving the subscribers to their current plan  I (IBC1=IBC2)&(IBP1=IBP2) W !!,\*7,"You must move the subscribers to a different plan!" G PROCQ  ;  ; - set name and plan number  S IBC1N=$P($G(^DIC(36,+IBC1,0)),U,1)  S IBP1N=$P($G(^IBA(355.3,+IBP1,0)),U,3,4),IBP1X=$P(IBP1N,U,2)  S IBP1X=$S(IBP1X]"":IBP1X,1:"<Not Specified>")  S IBP1N=$S($P(IBP1N,U,1)="":"<Not Specified>",1:$P(IBP1N,U,1))  S IBC2N=$P($G(^DIC(36,+IBC2,0)),U,1)  S IBP2N=$P($G(^IBA(355.3,+IBP2,0)),U,3,4),IBP2X=$P(IBP2N,U,2)  S IBP2X=$S(IBP2X]"":IBP2X,1:"<Not Specified>")  S IBP2N=$S($P(IBP2N,U,1)="":"<Not Specified>",1:$P(IBP2N,U,1))  ;  ; - ask if they want to delete the old insurance  S DIR(0)="Y",DIR("A")="Do you want to EXPIRE the old plan by entering the new plan Effective date"  S DIR("B")="NO"  S DIR("?")="If you wish to apply Effective Date, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I $D(DIRUT) G PROCQ  S IBSPLIT=''Y  ; if yes then  ; - ask the effective date of the new insurance  I IBSPLIT D  I IBQ G PROCQ  . S IBQ=0  . S %DT="AEX",%DT("A")="Effective Date of the new Plan: "  . W ! D ^%DT K %DT I Y'>0 S IBQ=1 Q  . S IBEFFDT=$P(+Y,".")  . S IBEXPDT=$$FMADD^XLFDT(IBEFFDT,-1)  ;  ; - ask are they sure  W !!!,"You selected to move ",IBSUB," subscribers and "  W $S(IBSPLIT:"EXPIRE",1:"REPLACE")," the old plan in the patient",!,"profile.",!  W !?5,"FROM Insurance Company ",IBC1N  W !?10,"Plan Name ",IBP1N," Number ",IBP1X  W !?5,"TO Insurance Company ",IBC2N  W !?10,"Plan Name ",IBP2N," Number ",IBP2X  I IBSPLIT D  . W !?5,"BY switching to the new Insurance/Plan"  . W !?10,"with Effective Date ",$$DAT2^IBOUTL(IBEFFDT)  W !  W !,"Please Note that the old insurance group plan will be "  W $S(IBSPLIT:"EXPIRED",1:"REPLACED")," in the patient",!,"profile!",!  ;  S DIR(0)="Y",DIR("A")="Okay to continue"  S DIR("?")="If you wish to move these subscribers, enter 'Yes' - otherwise, enter 'No.'"  W ! D ^DIR K DIR  I 'Y W !!,?10,"<Okay, nothing moved>" G PROCQ  ;  ; - should annual benefits be moved?  S (IBAB,IBQ)=0  I $D(^IBA(355.4,"APY",IBP1)),'$D(^IBA(355.4,"APY",IBP2)) D  G:IBQ PROCQ  .S DIR(0)="Y",DIR("A")="Okay to add "\_IBC1N\_"'s plan Annual Benefits to "\_IBC2N\_"'s plan"  .S DIR("?")="If you wish to move these Annual Benefits, enter 'Yes' - otherwise, enter 'No.'"  .W ! D ^DIR K DIR I $D(DIRUT)!$D(DIROUT)!$D(DUOUT)!$D(DTOUT) S IBQ=1  .S:Y IBAB=1 K DIRUT,DUOUT,DTOUT,DIROUT  ;  ; - copy annual benefits over to the new plan  I IBAB D  .S IBI=0 F  S IBI=$O(^IBA(355.4,"C",IBP1,IBI)) Q:'IBI  D  ..S IBIAB=$G(^IBA(355.4,IBI,0)) Q:'IBIAB  ..S X=+IBIAB,DIC(0)="L",DLAYGO=355.4,DIC="^IBA(355.4,"  ..K DD,DO D FILE^DICN Q:+Y<0 S IBCAB=+Y  ..S $P(^IBA(355.4,IBCAB,0),"^",2)=IBP2  ..S $P(^IBA(355.4,IBCAB,0),"^",5,6)=$P(IBIAB,"^",5,6)  ..F I=1:1:5 I $G(^IBA(355.4,IBI,I))]"" S ^IBA(355.4,IBCAB,I)=^(I)  ..S DA=IBCAB,DIK="^IBA(355.4," D IX1^DIK,EDUP^IBCNSA2  ;  ; - should plan comments be copied over to the new plan?  S (IBAB,IBQ)=0  I $P($G(^IBA(355.3,IBP1,11,0)),U,4),'$P($G(^IBA(355.3,IBP2,11,0)),U,4) D  G:IBQ PROCQ  .S DIR(0)="Y"  .S DIR("A")="Okay to add "\_IBC1N\_"'s Comments to "\_IBC2N\_"'s plan"  .S DIR("?")="If you wish to move these Comments, enter 'Yes'"  .S DIR("?")=DIR("?")\_" - otherwise, ente"  .W ! D ^DIR K DIR I $D(DIRUT)!$D(DIROUT)!$D(DUOUT)!$D(DTOUT) S IBQ=1  .S:Y IBAB=1 K DIRUT,DUOUT,DTOUT,DIROUT  ;  ; - copy plan comments over to the new plan  I IBAB D  .S DIC="^IBA(355.3,"\_IBP2\_",11,",DIC(0)="L",DIC("P")=355.311  .S IBI=0 F  S IBI=$O(^IBA(355.3,IBP1,11,IBI)) Q:'IBI  D  ..I $G(^IBA(355.3,IBP1,11,IBI,0))]"" S X=^(0) D FILE^DICN  ;  ; The MailMan bulletin header  D BHEAD^IBCNSUR3  ;  ; - move the subscribers to the new plan  W !!,"Moving subscribers "  S DFN=0 F  S DFN=$O(^TMP($J,"IBCNSUR",DFN)) Q:'DFN  D  .S IBCDFN=0 F  S IBCDFN=$O(^TMP($J,"IBCNSUR",DFN,IBCDFN)) Q:'IBCDFN  D  ..Q:$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)'=IBP1  ..;  ..D ADS^IBCNSUR3(DFN,IBCDFN)  ..I 'IBSPLIT D MODIFINS(IBC2,IBP2,DFN,IBCDFN) ;regular mode  ..I IBSPLIT D SPLITINS(IBC2,IBP2,DFN,IBCDFN,IBEFFDT,IBEXPDT)  ..; - merge previous benefits used  ..S IBDAT="" F  S IBDAT=$O(^IBA(355.5,"APPY",DFN,IBP1,IBDAT)) Q:IBDAT=""  D  ...S IBCDFN1=0 F  S IBCDFN1=$O(^IBA(355.5,"APPY",DFN,IBP1,IBDAT,IBCDFN1)) Q:'IBCDFN1  I IBCDFN1=IBCDFN S IBBU=$O(^(IBCDFN1,0)) D  ....I '$D(^IBA(355.4,"APY",IBP2,IBDAT)) D DBU^IBCNSJ(IBBU) Q  ....D MERG^IBCNSJ13(IBP2,IBBU)  ..;  ..W "."  ;  W !!,"Done. All subscribers were moved as requested!",!  D DONE^IBCNSUR3  W !,"The Bulletin was sent to you and members of 'IB NEW INSURANCE' Mail Group.",!  R !!,?10,"Press any key to continue. ",IBX:DTIME  ;  ; - finish processing in IBCNSUR (keep RSIZE down)  D PROC^IBCNSUR  ;  ; PROCQ ;I 'IBSTOP S IBQUIT=0 D ASK^IBCOMC2 I IBQUIT=1 S IBSTOP=1  K ^TMP($J,"IBCNSUR")  K ^TMP($J,"IBCNSUR1")  Q  ;  ; modify the ins plan MODIFINS(IBC2,IBP2,DFN,IBCDFN) ;  N IBXXX,DIE,DA,DR,IBX  ; - change the policy company  S IBXXX='$G(^DPT(DFN,.312,IBCDFN,1))  S DIE="^DPT(DFN,.312,",DA(1)=DFN,DA=IBCDFN,DR=".01///`"\_IBC2 D ^DIE K DIE,DA,DR  I IBXXX S $P(^DPT(DFN,.312,IBCDFN,1),"^",1,2)="^"  ;  ; - repoint Insurance Reviews to the new company  S IBX=0 F  S IBX=$O(^IBT(356.2,"D",DFN,IBX)) Q:'IBX  I $P($G(^IBT(356.2,IBX,1)),"^",5)=IBCDFN S DIE="^IBT(356.2,",DA=IBX,DR=".08////"\_IBC2 D ^DIE K DIE,DA,DR  ;  ; - change the policy plan  D SWPL^IBCNSJ13(IBP2,DFN,IBCDFN)  Q  ;  ;  ;  ; change the ins plan effective IBEFFDT SPLITINS(IBC2,IBP2,DFN,IBCDFN,IBEFFDT,IBEXPDT) ;  N IBX,IBZ,IBZ1,IBRT,IBI,IBIEN,IBCDFN2,IBERR,DIK,DA,DIE,DR,DGRUGA08  S IBZ=$G(^DPT(DFN,.312,IBCDFN,0))  S IBZ1=$G(^DPT(DFN,.312,IBCDFN,1))  ; - ignore if the old plan expired  I $P(IBZ,U,4),$P(IBZ,U,4)<IBEFFDT Q  ; - if the ins is effective later - no need to split  I $P(IBZ,U,8),$P(IBZ,U,8)'<IBEFFDT D MODIFINS(IBC2,IBP2,DFN,IBCDFN) Q  ;  S DGRUGA08=1 ; Disable HL7 triggered by 2.312/3 and 2.312/8  ; - create the new insurance record for the DFN (clone)  S IBI="+1,"\_DFN\_","  ; - add a record  S IBRT(2.312,IBI,.01)=IBC2  D UPDATE^DIE("","IBRT","IBIEN","IBERR")  I $D(IBERR) Q  ; error  I '$G(IBIEN(1)) Q  ; error  S IBCDFN2=+IBIEN(1)  ; - clone the insurance data  M ^DPT(DFN,.312,IBCDFN2)=^DPT(DFN,.312,IBCDFN)  S $P(^DPT(DFN,.312,IBCDFN2,0),U,1)=IBC2  S $P(^DPT(DFN,.312,IBCDFN2,0),U,8)=IBEFFDT  ; - now reindex  S DA(1)=DFN,DA=IBCDFN2,DIK="^DPT("\_DFN\_",.312,"  D IX1^DIK  ; - change the policy plan  D SWPL^IBCNSJ13(IBP2,DFN,IBCDFN2)  ; - set the expiration date  S $P(^DPT(DFN,.312,IBCDFN,0),U,4)=IBEXPDT  S DA(1)=DFN,DA=IBCDFN,DIK="^DPT("\_DFN\_",.312,"  D IX1^DIK  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| IBCNSUR1 ;ALB/CPM/CMS - MOVE SUBSCRIBERS TO DIFFERENT PLAN (CON'T)   ;;2.0;INTEGRATED BILLING;\*\*103,225,276\*\*;21-MAR-94  ;;Per VHA Directive 10-93-142, this routine should not be modified.  Q  ;  ; PROC ; - Top of processing from IBCNSUR  ; Move subscribers to another company's insurance plan.  N IBCNS,IBPLAN,IBC1,IBC1N,IBC1X,IBC2,IBC2N,IBC2X,IBCPOL  N IBP1,IBP1N,IBP1X,IBP2,IBP2N,IBP2X,IBQ,IBQUIT,IBSUB,DFN,IBCDFN  N IBXXX,IBX,IBDAT,IBCDFN1,IBNP,IBAB,IBI,IBIAB,IBCAB,IBW,IBST  N DIC,DIE,DR,DA,D0,DIR,DIRUT,DIROUT,DTOUT,DUOUT,I,X,Y,DIK,DLAYGO  N IBSPLIT,IBEFFDT,IBEXPDT,REF,IBLN  **N IBDEAD,IBEFDT,IBEFDT1,IBEFDT2,IBGRP,IBSUBACT,IBSUBID,IBVALUE**  ;  K ^TMP($J,"IBCNSUR") ; subscribers  K ^TMP($J,"IBCNSUR1") ; e-mail bulletin  S REF=$NA(^TMP($J,"IBCNSUR1")),IBLN=0  ;  S IBQUIT=0  W !!!,"=====================",!,"MOVE SUBSCRIBERS FROM",!,"====================="  W !!,"Select the Insurance Company and Plan to move subscribers FROM.",!  ;  ; - select company/plan for subscribers to be moved  S IBQUIT=0  D SEL^IBCNSUR(0)  I IBQUIT S IBSTOP=1 G PROCQ **; 2.6.16.1  ; - ask if they want to move the entire group plan  S DIR(0)="Y",DIR("A")="Do you want to move the entire group plan"  S DIR("B")="YES"  S DIR("?")="If you wish to select which subscribers are moved, enter 'No' - otherwise, enter 'Yes' and all subscribers will be moved"  W ! D ^DIR K DIR  I Y="^" S IBQUIT=1 G PROCQ  S IBGRP=Y  ;  I IBGRP G PROCB  ; if yes then proceed with collecting the subscribers for the entire plan.  ;  ; 2.6.16.2 if no, the proceed with the filtering questions.  ; - ask if they want to continue because they are about to select individual subscribers  S DIR(0)="Y",DIR("A")="You have selected to move individual subscribers. Okay to continue"  S DIR("B")=""  S DIR("?")="If you wish to continue selecting which subscribers are moved, enter 'Yes' - otherwise, enter 'No' to quit."  W ! D ^DIR K DIR  ; if yes then proceed with collecting the subscribers for the entire plan.  I ‘+Y!(Y="^") S IBQUIT=1 G PROCQ  ; QUIT  ;  ; - ask if they want to filter out Deceased Patients  S DIR(0)="Y",DIR("A")="Do you want to filter out deceased patients?"  S DIR("B")="YES"  S DIR("?")="If you wish to ignore Deceased Patients in the selection process, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I Y="^" S IBQUIT=1 G PROCQ  S IBDEAD=+Y   ; 1=ignore deceased patients, 0=include deceased patients.  ;  ; - ask if they want to filter based on Subscriber ID  S DIR(0)="Y",DIR("A")="Do you want to filter Subscriber ID?"  S DIR("B")="YES"  S DIR("?")="If you wish to filter subscribers based upon the Subscriber ID, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I Y="^" S IBQUIT=1 G PROCQ  S IBSUBID=+Y   ; 1=filter based upon the Subscriber ID, 0=ignore Subscriber IDs.  S IBQUIT=0  I +IBSUBID D  G:IBQUIT PROCQ  .;  .; - ask user to enter the value that subscriber IDs need to 'contain'  .S DIR(0)="FAO",DIR("A")="Filter Subscriber IDs that contain: "   .S DIR("?")="Enter the value that Subscriber IDs should contain. Typing NULL means blank values."   .D ^DIR K DIR  .I Y="^" S IBQUIT=1 Q  .S IBVALUE=Y  . Q  ;  ; - ask if they want to filter based on ACTIVE or INACTIVE  S DIR(0)="Y",DIR("A")="Do you want to filter active or inactive policies?"  S DIR("B")="YES"  S DIR("?")="If you wish to filter subscribers based upon ACTIVE or INACTIVE policy status, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I Y="^" S IBQUIT=1 G PROCQ  S IBSUBACT=+Y   ; 1=filter based upon the ACTIVE or INACTIVE, 0=ignore ACTIVE status.  ;  I IBSUBACT D  G:IBQUIT PROCQ  .; Filter based on Active or Inactive policies.  .S DIR(0)="SA^1:1. Active Policies;2:2. Inactive Policies"  .S DIR("A")=" SELECT WHICH POLICIES TO FILTER (1 OR 2): "  .S DIR("A",1)="1. Active Policies"  .S DIR("A",2)="2. Inactive Policies"  .D ^DIR K DIR I Y<0!$D(DIRUT) S IBQUIT=1 Q  .S IBSUBACT=Y K Y  .Q  ;  ; - ask if they want to filter based on Effective Dates  S DIR(0)="Y",DIR("A")="Do you want to filter Effective Dates?"  S DIR("B")="YES"  S DIR("?")="If you wish to filter subscribers based upon Effective Dates, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I Y="^" S IBQUIT=1 G PROCQ  S IBEFDT=+Y   ; 1=filter based upon Effective Dates, 0=ignore Effective Dates.  ; PROCA ; Enter Effective Date range to filter subscribers.  W ! S DIR(0)="DA",DIR("A")="Start with DATE: ",DIR("?")="Enter the earliest Effective Date to filter Subscribers."  D ^DIR I ‘Y S IBQUIT=1 G PROCQ  S IBEFDT1=Y  W ! S DIR(0)="DA",DIR("A")="Go to DATE: ",DIR("?")="Enter the latest Effective Date to filter Subscribers."  D ^DIR I ‘Y S IBQUIT=1 G PROCQ  S IBEFDT2=Y  I IBEFDT2<IBEFDT1 W !, "End date cannot be less than Start date…Please re-enter date range." G PROCA  ;  ;;; Need to enter a LISTMAN screen that will list the filtered subscribers based upon the above filters.  ;;; Upon the subscribers being selected that from the LISTMAN screen proceed with collecting the  ;;; plan subscribers.**  **PROCB ; - collect the plan subscribers**   ;  ; - collect the plan subscribers  S IBC1=IBCNS,IBP1=IBPLAN  W !!,"Collecting Subscribers ..." **I IBGRP S IBSUB=$$SUBS^IBCNSJ(IBC1,IBP1,0,"^TMP($J,""IBCNSUR"")")  I ‘IBGRP S IBSUB=$$SUBLKP^IBCNSUR4(IBDEAD,IBSUBID,IBVALUE,IBSUBACT,IBACTV,IBEFDT,IBEFDT1,IBEFDT2) ; This is a new sub-routine to collect the subscribers using the various filters.**  **;S IBSUB=$$SUBS^IBCNSJ(IBC1,IBP1,0,"^TMP($J,""IBCNSUR"")")**  I 'IBSUB W !!,?5,\*7,"\* This **group** plan has no subscribers!" S IBQUIT=1 G PROCQ  W !!,"This **group** plan has ",+IBSUB," subscribers. All subscribers will be moved."  ;   ; - select company/plan to move subscribers  W !!!,"MOVE SUBSCRIBERS TO"  W !!,"Select the Insurance Company and Plan to move subscribers TO.",!  D SEL^IBCNSUR(1)  I IBQUIT G PROCQ  I $P($G(^DIC(36,IBCNS,0)),"^",5) W !!,\*7,"You must move the subscribers to an active insurance company!" G PROCQ  S IBC2=IBCNS,IBP2=IBPLAN  ;  ; - make sure not moving the subscribers to their current plan  I (IBC1=IBC2)&(IBP1=IBP2) W !!,\*7,"You must move the subscribers to a different group plan!" G PROCQ  ;  ; - set name and plan number  S IBC1N=$P($G(^DIC(36,+IBC1,0)),U,1)  S IBP1N=$P($G(^IBA(355.3,+IBP1,0)),U,3,4),IBP1X=$P(IBP1N,U,2)  S IBP1X=$S(IBP1X]"":IBP1X,1:"<Not Specified>")  S IBP1N=$S($P(IBP1N,U,1)="":"<Not Specified>",1:$P(IBP1N,U,1))  S IBC2N=$P($G(^DIC(36,+IBC2,0)),U,1)  S IBP2N=$P($G(^IBA(355.3,+IBP2,0)),U,3,4),IBP2X=$P(IBP2N,U,2)  S IBP2X=$S(IBP2X]"":IBP2X,1:"<Not Specified>")  S IBP2N=$S($P(IBP2N,U,1)="":"<Not Specified>",1:$P(IBP2N,U,1))  ;  ; - ask if they want to delete the old insurance  S DIR(0)="Y",DIR("A")="Do you want to EXPIRE the old **patient policy(s) by** entering the new **policy** Effective date"   S DIR(0)="Y",DIR("A")="Do you want to EXPIRE the old **patient policy(s)** by entering the new **policy** Effective date"   S DIR("B")="NO"  S DIR("?")="If you wish to apply Effective Date, enter 'Yes' - otherwise, enter 'No'"  W ! D ^DIR K DIR  I $D(DIRUT) G PROCQ  S IBSPLIT=''Y  ; if yes then  ; - ask the effective date of the new insurance  I IBSPLIT D  I IBQ G PROCQ  . S IBQ=0  . S %DT="AEX",%DT("A")="Effective Date of the new **Policy(s)**: "  . W ! D ^%DT K %DT I Y'>0 S IBQ=1 Q  . S IBEFFDT=$P(+Y,".")  . S IBEXPDT=$$FMADD^XLFDT(IBEFFDT,-1)   ;  ; - ask are they sure  W !!!,"You selected to move ",IBSUB," subscribers and "  W $S(IBSPLIT:"EXPIRE",1:"REPLACE")," the old **policy** in the patient",!,"profile and load a new patient policy.",!  W !?5,"FROM Insurance Company ",IBC1N  W !?10,"Plan Name ",IBP1N," Number ",IBP1X  W !?5,"TO Insurance Company ",IBC2N  W !?10,"Plan Name ",IBP2N," Number ",IBP2X  I IBSPLIT D  . W !?5,"BY switching to the new Insurance/Plan"  . W !?10,"with Effective Date ",$$DAT2^IBOUTL(IBEFFDT)  W !  W !,"Please Note that the old insurance patient **policy** will be "  W $S(IBSPLIT:"EXPIRED",1:"REPLACED")," in the patient",!,"profile!",!  ;  S DIR(0)="Y",DIR("A")="Okay to continue"  S DIR("?")="If you wish to move these subscribers, enter 'Yes' - otherwise, enter 'No.'"  W ! D ^DIR K DIR  I 'Y W !!,?10,"<Okay, nothing moved>" G PROCQ  ;  ; - should annual benefits be moved?  S (IBAB,IBQ)=0  I $D(^IBA(355.4,"APY",IBP1)),'$D(^IBA(355.4,"APY",IBP2)) D  G:IBQ PROCQ  .S DIR(0)="Y",DIR("A")="Okay to add "\_IBC1N\_"'s plan Annual Benefits to "\_IBC2N\_"'s plan"  .S DIR("?")="If you wish to move these Annual Benefits, enter 'Yes' - otherwise, enter 'No.'"  .W ! D ^DIR K DIR I $D(DIRUT)!$D(DIROUT)!$D(DUOUT)!$D(DTOUT) S IBQ=1  .S:Y IBAB=1 K DIRUT,DUOUT,DTOUT,DIROUT  ;  ; - copy annual benefits over to the new plan  I IBAB D  .S IBI=0 F  S IBI=$O(^IBA(355.4,"C",IBP1,IBI)) Q:'IBI  D  ..S IBIAB=$G(^IBA(355.4,IBI,0)) Q:'IBIAB  ..S X=+IBIAB,DIC(0)="L",DLAYGO=355.4,DIC="^IBA(355.4,"  ..K DD,DO D FILE^DICN Q:+Y<0 S IBCAB=+Y  ..S $P(^IBA(355.4,IBCAB,0),"^",2)=IBP2  ..S $P(^IBA(355.4,IBCAB,0),"^",5,6)=$P(IBIAB,"^",5,6)  ..F I=1:1:5 I $G(^IBA(355.4,IBI,I))]"" S ^IBA(355.4,IBCAB,I)=^(I)  ..S DA=IBCAB,DIK="^IBA(355.4," D IX1^DIK,EDUP^IBCNSA2  ;  ; - should plan comments be copied over to the new plan?  S (IBAB,IBQ)=0  I $P($G(^IBA(355.3,IBP1,11,0)),U,4),'$P($G(^IBA(355.3,IBP2,11,0)),U,4) D  G:IBQ PROCQ  .S DIR(0)="Y"  .S DIR("A")="Okay to add "\_IBC1N\_"'s Comments to "\_IBC2N\_"'s plan"  .S DIR("?")="If you wish to move these Comments, enter 'Yes'"  .S DIR("?")=DIR("?")\_" - otherwise, ente"  .W ! D ^DIR K DIR I $D(DIRUT)!$D(DIROUT)!$D(DUOUT)!$D(DTOUT) S IBQ=1  .S:Y IBAB=1 K DIRUT,DUOUT,DTOUT,DIROUT  ;  ; - copy plan comments over to the new plan  I IBAB D  .S DIC="^IBA(355.3,"\_IBP2\_",11,",DIC(0)="L",DIC("P")=355.311  .S IBI=0 F  S IBI=$O(^IBA(355.3,IBP1,11,IBI)) Q:'IBI  D  ..I $G(^IBA(355.3,IBP1,11,IBI,0))]"" S X=^(0) D FILE^DICN  ;  ; The MailMan bulletin header  D BHEAD^IBCNSUR3  ;  ; - move the subscribers to the new plan  W !!,"Moving subscribers "  S DFN=0 F  S DFN=$O(^TMP($J,"IBCNSUR",DFN)) Q:'DFN  D  .S IBCDFN=0 F  S IBCDFN=$O(^TMP($J,"IBCNSUR",DFN,IBCDFN)) Q:'IBCDFN  D  ..Q:$P($G(^DPT(DFN,.312,IBCDFN,0)),"^",18)'=IBP1  ..;  ..D ADS^IBCNSUR3(DFN,IBCDFN)  ..I 'IBSPLIT D MODIFINS(IBC2,IBP2,DFN,IBCDFN) ;regular mode  ..I IBSPLIT D SPLITINS(IBC2,IBP2,DFN,IBCDFN,IBEFFDT,IBEXPDT)  ..; - merge previous benefits used  ..S IBDAT="" F  S IBDAT=$O(^IBA(355.5,"APPY",DFN,IBP1,IBDAT)) Q:IBDAT=""  D  ...S IBCDFN1=0 F  S IBCDFN1=$O(^IBA(355.5,"APPY",DFN,IBP1,IBDAT,IBCDFN1)) Q:'IBCDFN1  I IBCDFN1=IBCDFN S IBBU=$O(^(IBCDFN1,0)) D  ....I '$D(^IBA(355.4,"APY",IBP2,IBDAT)) D DBU^IBCNSJ(IBBU) Q  ....D MERG^IBCNSJ13(IBP2,IBBU)  ..;  ..W "."  ;  W !!,"Done. All subscribers were moved as requested!",!  D DONE^IBCNSUR3  W !,"The Bulletin was sent to you and members of 'IB NEW INSURANCE' Mail Group.",!  R !!,?10,"Press any key to continue. ",IBX:DTIME  ;  ; - finish processing in IBCNSUR (keep RSIZE down)  D PROC^IBCNSUR  ;  ; PROCQ ;I 'IBSTOP S IBQUIT=0 D ASK^IBCOMC2 I IBQUIT=1 S IBSTOP=1  K ^TMP($J,"IBCNSUR")  K ^TMP($J,"IBCNSUR1")  Q  ;  ; modify the ins plan MODIFINS(IBC2,IBP2,DFN,IBCDFN) ;  N IBXXX,DIE,DA,DR,IBX  ; - change the policy company  S IBXXX='$G(^DPT(DFN,.312,IBCDFN,1))  S DIE="^DPT(DFN,.312,",DA(1)=DFN,DA=IBCDFN,DR=".01///`"\_IBC2 D ^DIE K DIE,DA,DR  I IBXXX S $P(^DPT(DFN,.312,IBCDFN,1),"^",1,2)="^"  ;  ; - repoint Insurance Reviews to the new company  S IBX=0 F  S IBX=$O(^IBT(356.2,"D",DFN,IBX)) Q:'IBX  I $P($G(^IBT(356.2,IBX,1)),"^",5)=IBCDFN S DIE="^IBT(356.2,",DA=IBX,DR=".08////"\_IBC2 D ^DIE K DIE,DA,DR  ;  ; - change the policy plan  D SWPL^IBCNSJ13(IBP2,DFN,IBCDFN)  Q  ;  ;  ;  ; change the ins plan effective IBEFFDT SPLITINS(IBC2,IBP2,DFN,IBCDFN,IBEFFDT,IBEXPDT) ;  N IBX,IBZ,IBZ1,IBRT,IBI,IBIEN,IBCDFN2,IBERR,DIK,DA,DIE,DR,DGRUGA08  S IBZ=$G(^DPT(DFN,.312,IBCDFN,0))  S IBZ1=$G(^DPT(DFN,.312,IBCDFN,1))  ; - ignore if the old plan expired  I $P(IBZ,U,4),$P(IBZ,U,4)<IBEFFDT Q  ; - if the ins is effective later - no need to split  I $P(IBZ,U,8),$P(IBZ,U,8)'<IBEFFDT D MODIFINS(IBC2,IBP2,DFN,IBCDFN) Q  ;  S DGRUGA08=1 ; Disable HL7 triggered by 2.312/3 and 2.312/8  ; - create the new insurance record for the DFN (clone)  S IBI="+1,"\_DFN\_","  ; - add a record  S IBRT(2.312,IBI,.01)=IBC2  D UPDATE^DIE("","IBRT","IBIEN","IBERR")  I $D(IBERR) Q  ; error  I '$G(IBIEN(1)) Q  ; error  S IBCDFN2=+IBIEN(1)  ; - clone the insurance data  M ^DPT(DFN,.312,IBCDFN2)=^DPT(DFN,.312,IBCDFN)  S $P(^DPT(DFN,.312,IBCDFN2,0),U,1)=IBC2  S $P(^DPT(DFN,.312,IBCDFN2,0),U,8)=IBEFFDT  ; - now reindex  S DA(1)=DFN,DA=IBCDFN2,DIK="^DPT("\_DFN\_",.312,"  D IX1^DIK  ; - change the policy plan  D SWPL^IBCNSJ13(IBP2,DFN,IBCDFN2)  ; - set the expiration date  S $P(^DPT(DFN,.312,IBCDFN,0),U,4)=IBEXPDT  S DA(1)=DFN,DA=IBCDFN,DIK="^DPT("\_DFN\_",.312,"  D IX1^DIK  Q |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSUR4** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.16.3, 2.6.16.4, 2.6.16.6 | | | | | | | | |
| **Related Options** | Move Subscribers to a Different Plan [IBCN MOVE SUBSCRIB TO PLAN] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCNSUR1 | | | | | None | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Current Logic** | | | | | | | | | |
| N/A | | | | | | | | | |
| **Modified Logic (Changes are in bold)** | | | | | | | | | |
| **IBCNSUR4 ;ALB/NLR - SELECTIVE SUBSCRIBER LOOK-UP UTILITY ; 14-APR-15  ;;2.0;INTEGRATED BILLING;\*\*549\*\*;14-APR-15;Build 120  ;;Per VHA Directive 2004-038, this routine should not be modified.  ; SUBLKP(IBDD,IBSBID,IBVAL,IBSBACT,IBEFDT,IBEFDT1,IBEFDT2) ; Look-up Utility for Selective Subscribers  ; Input: IBDD -- Deceased Subscribers Indicator (1 - Include  ; Deceased, 0 - Ignore Deceased)  ; IBSBID -- Subscriber ID Filter (1 - Use IBVAL to filter  ; Subscriber IDs, 0 - Ignore Subscriber IDs)  ; IBVAL -- Use the contained value to screen Subscriber IDs.  ; IBSBACT -- Subscriber Filter for Active Indicator (0 - Ignore  ; Active Status, 1 - Filter Active, 2 - Filter Inactive)  ; to be excluded from selection  ; IBEFDT -- Effective Date Filter Indicator (1 - Use Effective  ; Dates as a filter, 0 - Ignore Effective Dates.)  ; IBEFDT1 -- Effective Date Filter Start Date.  ; IBEFDT2 -- Effective Date Filter End Date.  ;  ; Output: IBCNT -- Number of Subscriber Policies to Move.  ;  ; a.) Need to create a LISTMAN list of Filtered Subscribers to Move to a Different Plan.  ; b.) The ListMan Title will be "Patient Policy Lookup" (REQ # 2.6.16.3)  ; c.) It will count the number of subscribers that met all of the filtering requirements. (REQ # 2.6.16.3)  ; d.) It will display the Patient Name, last 4 of SSN, DOB, Subscriber ID, Effective Date,  ; Expiration Date, Patient ID. (REQ # 2.6.16.3)  ; e.) The ListMan list will need to be able to allow a Search capability (REQ # 2.6.16.4); - It will also allow the user to select one or more subscribers to be moved (including  ; a range, i.e., 1,3-5,7) (REQ # 2.6.16.6)  Q  ;** | | | | | | | | | |

| **Routines** | **Activities** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Routine Name** | **IBCNSU2** | | | | | | | | |
| **Enhancement Category** | New | | Modify | Delete | | | No Change | | |
| **RTM** | 2.6.16.5 | | | | | | | | |
| **Related Options** | Move Subscribers to a Different Plan [IBCN MOVE SUBSCRIB TO PLAN] | | | | | | | | |
| **Related Routines** | **Routines “Called By”** | | | | | **Routines “Called”** | | | |
|  | IBCMDT  IBCNSJ14  IBCNSJ5  IBCNSM31  IBCNSUR  IBCOPP | | | | | YN^IBCNSM  OK^IBCNSM3  EXPAND^IBTRE | | | |
| **Data Dictionary (DD) References** | None | | | | | | | | |
| **Related Protocols** | None | | | | | | | | |
| **Related Integration Control Registrations (ICRs)** | None | | | | | | | | |
| **Data Passing** | Input | Output Reference | | | Both | | | Global Reference | Local |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | | | | | |

| **Current Logic** |
| --- |
| .  .  .   ; LKP(IBCNS,IBIND,IBMULT,IBSEL,IBALR,IBW) ; Look-up Utility for Insurance Plans  ; Input: IBCNS -- Pointer to the ins. company in file #36  ; IBIND -- Include Individual Plans? (1 - Yes | 0 - No)  ; IBMULT -- If set to 1, allows multiple plans to be chosen  ; IBALR -- May be set to point to plan in file #355.3  ; to be excluded from selection  ; IBW -- If set to 1, allows inactive plans to be chosen  ; Output: IBSEL -- Set to the pointer to the plan in file #355.3  ; if only a single plan is to be selected.  ;  ; The array ^TMP($J,"IBSEL",ptr)="" is returned  ; (where 'ptr' points to the plan in file  ; #355.3) if multiple plans are to be selected.  ;  I '$G(IBCNS) G LKPQ  N VALMY,VALMHDR  S IBIND=$G(IBIND)>0,IBW=$G(IBW)>0,IBMULT=+$G(IBMULT),IBSEL=0  D EN^VALM("IBCNS PLAN LOOKUP") LKPQ Q  ;.  . |

| **Modified Logic (Changes are in bold)** |
| --- |
| .  .  .   ;  ; LKP(IBCNS,IBIND,IBMULT,IBSEL,IBALR,IBW**,IBTLE**) ;**2.6.16.5** Look-up Utility for Insurance Plans  ; Input: IBCNS -- Pointer to the ins. company in file #36  ; IBIND -- Include Individual Plans? (1 - Yes | 0 - No)  ; IBMULT -- If set to 1, allows multiple plans to be chosen  ; IBALR -- May be set to point to plan in file #355.3  ; to be excluded from selection  ; IBW -- If set to 1, allows inactive plans to be chosen **; IBTLE – If set, then change the variable VALM(“TITLE”) to contain the value of IBTLE 2.6.16.5**   ; Output: IBSEL -- Set to the pointer to the plan in file #355.3  ; if only a single plan is to be selected.  ;  ; The array ^TMP($J,"IBSEL",ptr)="" is returned  ; (where 'ptr' points to the plan in file  ; #355.3) if multiple plans are to be selected.  ;  I '$G(IBCNS) G LKPQ  N VALMY,VALMHDR  S IBIND=$G(IBIND)>0,IBW=$G(IBW)>0,IBMULT=+$G(IBMULT),IBSEL=0  **I $G(IBTLE)’=”” S VALM(“TITLE”)=IBTLE ; 2.6.16.5**  D EN^VALM("IBCNS PLAN LOOKUP") LKPQ Q  .  . |

## Network Detailed Design

This modification does not include any changes to the network design.

## Service Oriented Architecture / ESS Detailed Design

This modification does not include any changes to the service oriented architecture/ESS design; therefore, all subsections below this do not apply to this effort.

### Service Description for <Consumed Service Name>

This modification does not include any changes to the service oriented architecture/ESS design.

### Service Design for <Provided Service Name>

This modification does not include any changes to the service oriented architecture/ESS design.

#### Introduction

##### Purpose and Scope of Service

This modification does not include any changes to services.

##### Links to Other Documents

Documents related to this effort are listed in References (Section 1.8).

#### Service Details

##### Service Identification

This modification does not include any changes to services.

##### Service Versions

This modification does not include any changes to services.

##### Summary of Design and Platform Details

###### SOA Pattern(s) Implemented

This modification does not include any changes to the platform.

###### COTS Platform vendor names and versions for hosting platform

This modification does not include any changes to the platform.

#### Dependencies

Refer to Software Interfaces (Section 6.2.1.1.3) for a list of software that must be installed in VistA prior to the release of this product.

#### Service Design Details

This modification does not include any changes to services; therefore, none of the subsections under this group apply to this effort.

##### Interface Technical Specs

This modification does not include any changes to services.

###### Service Invocation Type

This modification does not include any changes to services.

###### Service Interface Type

This modification does not include any changes to services.

###### Service Name

This modification does not include any changes to services.

###### Interface

This modification does not include any changes to services.

###### End Points

This modification does not include any changes to services.

###### Operations or Methods

This modification does not include any changes to services.

###### Message Schemas

This modification does not include any changes to services.

##### Information Model

This modification does not include any changes to services.

###### Class Diagram and Description of Entities Involved

This modification does not include any changes to services.

###### Mappings from ELDM to Standards Based Schemas

This modification does not include any changes to services.

##### Behavior Model (AKA Use Case Realization)

This modification does not include any changes to services.

###### Use Cases (Use Case Model)

This modification does not include any changes to services.

###### Interaction Diagrams

This modification does not include any changes to services.

#### Gap Analysis

This modification does not require a gap analysis.

##### Variances from Enterprise Target Architecture

This modification does not include any changes to the enterprise target architecture.

##### Variances from SLDs

This modification does not include any changes to the SLDs.

##### Variances from Standards and Policies

This modification does not include any variances from the standards and policies.

##### Justification for Exceptions and Mitigation

This modification does not include any exceptions.

# External System Interface Design

## Interface Architecture

This modification does not include any changes to the existing interface architecture.

## Interface Detailed Design

This modification does not include any new interface to/from the VistA system.

# Human-Machine Interface

This project does not change the human-machine interface, which is done via the VistA user options and ListMan screens.

## Interface Design Rules

This modification does not include any changes to the human-machine interface.

## Inputs

This modification does not include any changes to the human-machine interface.

## Outputs

This modification does not include any changes to the human-machine interface.

## Navigation Hierarchy

This modification does not include any changes to the navigation hierarchy.

### Screen [x.1]

This modification does not include any changes to the navigation hierarchy.

# Security and Privacy

## Security

This modification does not include any changes to VistA security.

## Privacy

This modification does not include any changes to VistA privacy.

Attachment A – Approval Signatures

This section is used to document the approval of the System Design Document. The review should be conducted face to face where signatures can be obtained ‘live’ during the review. If unable to conduct a face-to-face meeting then it should be held via LiveMeeting and concurrence captured during the meeting. The Scribe should add /es/name by each position cited. Example provided below.

The Chair of the governing Integrated Project Team (IPT), Business Sponsor, IT Program Manager, and Project Manager are required to sign.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Integrated Project Team (IPT) Chair >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Business Sponsor >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< IT Program Manager >

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signed: Date:

< Project Manager >

1. Additional Information
   1. RTM

The RTM is a separate document.

* 1. Packaging and Installation

Software packaging and installation will be done using the VistA Kernel Installation and Distribution System (KIDS) application.

* 1. Design Metrics

No special design metrics were required for this project.

* 1. Acronym List and Glossary

Table 59: Glossary

| Term | Meaning |
| --- | --- |
| ANSI | American National Standards Institute |
| API | Application Programming Interface |
| AR | Accounts Receivable |
| BRD | Business Requirements Document |
| CBO | Chief Business Office |
| CBOC | Community Based Outpatient Clinic |
| CLC | Community Living Center |
| CPAC | Consolidated Patient Account Centers |
| CPRS | Computerized Patient Record System |
| CT | Claims Tracking |
| DD | Data Dictionary |
| DG | Registration Module |
| DOD | Date of Death |
| EC | Eligibility Communicator |
| EDI | Electronic Data Interchange |
| EHR | Electronic Health Record |
| eIV | Electronic Insurance Verification |
| FSC | Financial Service Center |
| GUI | Graphical User Interface |
| HHS | Health and Human Services |
| HIPAA | Health Insurance Portability and Accountability Act |
| HL7 | Health Level 7 |
| HPID | Health Plan Identifier |
| IB | Integrated Billing |
| ICB | Insurance Capture Buffer |
| ICD | Interface Control Document |
| ISO | International Organization for Standardization |
| IM/IT | Information Management/Information Technology |
| IRM | Information Resource Manager |
| IVP | Insurance Verification Processor |
| KIDS | Kernel Installation and Distribution System |
| MCCF | Medical Care Collections Fund |
| MCCR | Medical Care Cost Recovery |
| MDWS | Medical Domain Web Services |
| NIF | National Insurance File |
| OED | Office of Enterprise Development |
| OEID | Other Entity Identifier |
| OIG | Office of Inspector General |
| OI&T | Office of Information and Technology |
| PD | Product Development |
| PICM | Patient Information Collection Management |
| PMO | Program Management Office |
| POC | Point of Care |
| PPACA | Patient Protection and Affordable Care Act |
| PS | Product Support |
| RPC | Remote Procedure Call |
| RSD | Requirements Specification Document |
| RUR | Revenue Utilization Review |
| SAC/SACC | Standards and Conventions/Standards and Conventions Committee |
| SDD | System Design Document |
| SQA | Software Quality Assurance |
| TPJI | Third Party Joint Inquiry |
| VA | Department of Veterans Affairs |
| VAMC | VA Medical Center |
| VHA | Veterans Health Administration |
| VISN | Veterans Integrated Service Network |
| VistA | Veterans Health Information Systems and Technology Architecture |

* 1. Required Technical Documents

Required technical documents are included in References (Section 1.8).

* 1. Attach Documents

There are no attached documents.

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| January 2015 | 2.8 | Updated to latest Section 508 guidelines and remediated with Common Look Office Tool | Process Management |
| September 2014 | 2.7 | Adds Enterprise Shared Services terms and requires AERB Compliance Certificate attachment. | Process Management |
| August 2014 | 2.6 | Signature block update authorized by AERB CR\_018934 | Process Management |
| March 2014 | 2.5 | Section 508 repairs to new version approved by AERB Chair approved | Process Management |
| August 2013 | 2.3 | Replaced the Service Architecture sub-section with new sub-sections for consumed and provided services. Also applied miscellaneous feedback from VA team. | ASD Enterprise Shared Services (ESS) Work Group |
| June 2013 | 1.3 | Upgraded to MS Office 2007-2010 format | Process Management |
| June 2013 | 1.2 | Address inconsistencies in Section 3, Conceptual Design, Correct headings | Process Management |
| March 2013 | 1.1 | Formatted to documentation standards and edited for Section 508 conformance | Process Management |
| January 2013 | 1.0 | Initial Document | PMAS Business Office |

Place latest revisions at top of table.

The Template Revision History pertains only to the format of the template. It does not apply to the content of the document or any changes or updates to the content of the document after distribution.

The Template Revision History can be removed at the discretion of the author of the document.

Remove blank rows.

See TOGAF® 9.1, Part III: ADM Guidelines & Techniques, Gap Analysis on TOGAF website at <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html>

1. Formerly known as the eIV Patient Insurance Update report [↑](#footnote-ref-2)
2. Formerly known as the eIV Patient Insurance Update report [↑](#footnote-ref-3)