Department of Veterans Affairs

Fee-Revenue Enhancements

System Design Document

IB\*2.0\*554

FB\*3.5\*163



February 2016

Version 0.06

Revision History

| Date | Version | Description | Author | Reviewers | Issue Date | Formal Review |
| --- | --- | --- | --- | --- | --- | --- |
| 07/07/2015 | 0.01 | Sections 1-4 for AERB Review Request | CPAC PMOSS | xxxx |  |  |
| 08/14/2015 | 0.02 | Initial Draft | xxxxx | xxxxx | 08/14/2015 | 08/21/2015 |
| 08/27/2015 | 0.03 | Updated based on VA Comments  Added Functionality bullets in section 2.3.1  Added diagrams in sections 3.1.1., 3.1.2, and 3.2.1 | xxxx | xxxx | 09/03/2015 |  |
| 10/20/2015 | 0.04 | Updates to Sections 5 & 6 | xxxx | xxxx |  |  |
| 1/19/2016 | 0.05 | Updated Table 2 Updates to Sections 5 & 6 | xxxx |  |  |  |
| 2/7/2016 | 0.06 | Updated per VA review comments | xxxx | xxxx |  |  |

| Activity | New Capability (1) | Feature Enhancement (2) |
| --- | --- | --- |
| **Field Deployment (A)** | Yes | Yes |
| **Cloud/Web Deployment (B)** | No | No |
| **Mobile Application (C)** | No | No |

Table of Contents

[1. Introduction 7](#_Toc442880844)

[1.1. Scope 7](#_Toc442880845)

[1.2. User Profiles 7](#_Toc442880846)

[2. Background 7](#_Toc442880847)

[2.1. Overview of the System 7](#_Toc442880848)

[2.2. Overview of the Business Process 8](#_Toc442880849)

[2.3. Overview of the Significant Requirements 8](#_Toc442880850)

[2.3.1. Overview of Functional Requirements 8](#_Toc442880851)

[2.3.2. Overview of Functional Workload/Performance Requirements 13](#_Toc442880852)

[2.3.3. Overview of Operational Requirements 13](#_Toc442880853)

[2.3.4. Overview of Technical Requirements 13](#_Toc442880854)

[2.3.5. Overview of Security or Privacy Requirements 13](#_Toc442880855)

[2.3.6. Overview of System Criticality and High Availability Requirements 13](#_Toc442880856)

[2.3.7. Single Sign-on Requirements 14](#_Toc442880857)

[2.3.8. Requirement for Use of Enterprise Portals 14](#_Toc442880858)

[2.3.9. Special Device Requirements 14](#_Toc442880859)

[3. Conceptual Design 14](#_Toc442880860)

[3.1. Conceptual Application Design 14](#_Toc442880861)

[3.1.1. Application Context 14](#_Toc442880862)

[3.1.2. High-Level Application Design 14](#_Toc442880863)

[3.1.3. Application Locations 17](#_Toc442880864)

[3.2. Conceptual Data Design 17](#_Toc442880865)

[3.2.1. Project Conceptual Data Model 17](#_Toc442880866)

[3.2.2. Database Information 18](#_Toc442880867)

[3.2.3. User Interface Data Mapping 18](#_Toc442880868)

[3.3. Conceptual Infrastructure Design 18](#_Toc442880869)

[3.3.1. System Criticality and High Availability 18](#_Toc442880870)

[3.3.2. Special Technology 19](#_Toc442880871)

[3.3.3. Technology Locations 19](#_Toc442880872)

[3.3.4. Conceptual Infrastructure Diagram 19](#_Toc442880873)

[3.3.4.1. Location of Environments and External Interfaces 19](#_Toc442880874)

[3.3.4.2. Conceptual Production String Diagram 19](#_Toc442880875)

[4. System Architecture 19](#_Toc442880876)

[4.1. Hardware Architecture 19](#_Toc442880877)

[4.2. Software Architecture 19](#_Toc442880878)

[4.3. Network Architecture 21](#_Toc442880879)

[4.4. SOA/ESS 22](#_Toc442880880)

[4.5. Enterprise Architecture 22](#_Toc442880881)

[5. Data Design 22](#_Toc442880882)

[5.1. Database Management System (DBMS) Files 22](#_Toc442880883)

[5.2. Non-DBMS Files 31](#_Toc442880884)

[5.3. Data View 31](#_Toc442880885)

[6. Detailed Design 31](#_Toc442880886)

[6.1. Hardware Detailed Design 31](#_Toc442880887)

[6.2. Software Detailed Design 31](#_Toc442880888)

[6.2.1. Conceptual Design 31](#_Toc442880889)

[6.2.1.1. Product Perspective 31](#_Toc442880890)

[6.2.1.1.1. User Interfaces 31](#_Toc442880891)

[6.2.1.1.2. Hardware Interfaces 32](#_Toc442880892)

[6.2.1.1.3. Software Interfaces 32](#_Toc442880893)

[6.2.1.1.4. Communications Interfaces 32](#_Toc442880894)

[6.2.1.1.5. Memory Constraints 32](#_Toc442880895)

[6.2.1.1.6. Special Operations 32](#_Toc442880896)

[6.2.1.2. Product Features 32](#_Toc442880897)

[6.2.1.3. User Characteristics 32](#_Toc442880898)

[6.2.1.4. Dependencies and Constraints 32](#_Toc442880899)

[6.2.2. Specific Requirements 32](#_Toc442880900)

[6.2.2.1. Database Repository 32](#_Toc442880901)

[6.2.2.2. System Features 32](#_Toc442880902)

[6.2.2.3. Nightly Process #1 36](#_Toc442880903)

[6.2.2.4. Billing Worklist 36](#_Toc442880904)

[6.2.2.5. Design Element Tables 38](#_Toc442880905)

[6.2.2.5.1. Routines (Entry Points) 38](#_Toc442880906)

[6.2.2.8.2. Templates 77](#_Toc442880907)

[6.2.2.8.3. Data Entries Affected by the Design 79](#_Toc442880908)

[6.2.2.8.4. Unique Record(s) 79](#_Toc442880909)

[6.2.2.8.5. File or Global Size Changes 79](#_Toc442880910)

[6.2.2.8.6. Mail Groups 79](#_Toc442880911)

[6.2.2.8.7. Security Keys 80](#_Toc442880912)

[6.2.2.8.8. Options 81](#_Toc442880913)

[6.2.2.8.9. Protocols 82](#_Toc442880914)

[6.2.2.8.10. RPC 83](#_Toc442880915)

[6.2.2.8.11. Constants Defined in Interface 83](#_Toc442880916)

[6.2.2.8.12. Variables Defined in Interface 84](#_Toc442880917)

[6.2.2.8.13. Types Defined in Interface 84](#_Toc442880918)

[6.2.2.8.14. GUI 84](#_Toc442880919)

[6.2.2.8.15. GUI Classes 84](#_Toc442880920)

[6.2.2.8.16. Current Form 84](#_Toc442880921)

[6.2.2.8.17. Modified Form 84](#_Toc442880922)

[6.2.2.8.18. Components on Form 84](#_Toc442880923)

[6.2.2.8.19. Events 84](#_Toc442880924)

[6.2.2.8.20. Methods 85](#_Toc442880925)

[6.2.2.8.21. Special References 85](#_Toc442880926)

[6.2.2.8.22. Class Events 85](#_Toc442880927)

[6.2.2.8.23. Class Methods 85](#_Toc442880928)

[6.2.2.8.24. Class Properties 85](#_Toc442880929)

[6.2.2.8.25. Uses Clause 85](#_Toc442880930)

[6.2.2.8.26. Forms 85](#_Toc442880931)

[6.2.2.8.27. Functions 86](#_Toc442880932)

[6.2.2.8.28. Dialog 87](#_Toc442880933)

[6.2.2.8.29. Help Frame 87](#_Toc442880934)

[6.2.2.8.30. HL7 Application Parameter 87](#_Toc442880935)

[6.2.2.8.31. HL7 Logical Link 88](#_Toc442880936)

[6.2.2.8.32. COTS Interface 88](#_Toc442880937)

[6.3. Network Detailed Design 88](#_Toc442880938)

[6.4. Security and Privacy 88](#_Toc442880939)

[6.4.1. Security 88](#_Toc442880940)

[6.4.2. Privacy 89](#_Toc442880941)

[6.5. SOA/ESS Detailed Design 89](#_Toc442880942)

[7. External System Interface Design 89](#_Toc442880943)

[8. Human-Machine Interface 89](#_Toc442880944)

[Attachment A – Approval Signatures 90](#_Toc442880945)

[Attachment B – Acronyms 91](#_Toc442880946)

[Attachment C – Additional Information 93](#_Toc442880947)

[A.1 Identification of Technology and Standards 93](#_Toc442880948)

[A.2 Constraining Policies, Directives and Procedures 93](#_Toc442880949)

[A.3 RTM 93](#_Toc442880950)

[A.4 Packaging and Installation 93](#_Toc442880951)

[A.5 Design Metrics 93](#_Toc442880952)

[Attachment D – Change Log 94](#_Toc442880953)

List of Tables

[Table 1: Primary and Secondary Users 7](#_Toc442880954)

[Table 2 : Functional Requirements 9](#_Toc442880955)

[Table 3 : Database Information 18](#_Toc442880956)

[Table 4 : FileMan Databases 22](#_Toc442880957)

[Table 5 : Functional Requirements (RSD) vs. Design Elements 33](#_Toc442880958)

[Table 6: (Grouping): Routines 38](#_Toc442880959)

[Table 7: Security Keys 80](#_Toc442880960)

[Table 8: Options 81](#_Toc442880961)

[Table 9: Protocols 82](#_Toc442880962)

[Table 10: RPCs 83](#_Toc442880963)

[Table 11: Constants Defined in Interface 83](#_Toc442880964)

[Table 12: Variables Defined in Interface 84](#_Toc442880965)

[Table 13: Types Defined in Interface 84](#_Toc442880966)

[Table 14: GUI 84](#_Toc442880967)

[Table 15: GUI Classes 84](#_Toc442880968)

[Table 16: Components on Form 84](#_Toc442880969)

[Table 17: Events 84](#_Toc442880970)

[Table 18: Methods 85](#_Toc442880971)

[Table 19: Special References 85](#_Toc442880972)

[Table 20: Class Events 85](#_Toc442880973)

[Table 21: Class Methods 85](#_Toc442880974)

[Table 22: Class Properties 85](#_Toc442880975)

[Table 23: Forms 85](#_Toc442880976)

[Table 24: Functions 86](#_Toc442880977)

[Table 25: Dialog 87](#_Toc442880978)

[Table 26: Help Frame 87](#_Toc442880979)

[Table 27: HL7 Application Parameter 87](#_Toc442880980)

[Table 28: HL7 Logical Link 88](#_Toc442880981)

[Table 29: COTS Interface 88](#_Toc442880982)

[Table 30: Acronym List 91](#_Toc442880983)

List of Figures

[Figure 1: Fee Revenue Enhancement Application Context Diagram 14](#_Toc442880984)

[Figure 2: Overview of EDI X12 Messaging 15](#_Toc442880985)

[Figure 3: Overview of Claims Processing and Eligibility (CP&E) System and Interfaces 16](#_Toc442880986)

[Figure 4: Data Model 17](#_Toc442880987)

Figure 5: Network Architecture…………………………………………………………………...………21

Figure 6: Enterprise Architecture………………………………………………………………………….22

# Introduction

The Veterans Health Administration (VHA) Chief Business Office (CBO), Purchased Care Program Office affirms that the current Veterans Health Information Systems and Technology Architecture (VistA) does not support the business processes for securing potential cost recovery of billable Non-Department of Veterans Affairs (VA) episodes of care in a timely and efficient manner. They are seeking enhancements to the VistA Fee Basis (FB) and Integrated Billing (IB)/Revenue software applications to automate the movement of billable Non-VA Care (NVC) cases from VistA FB through the Revenue IB software product, to include prioritization of high cost claims, automation of Veterans’ co-pay requirements and decision support for assessing clinical review of cases.

## Scope

The Fee-Revenue Enhancements Business Requirements Document (BRD) is available via Technical Services Project Repository (TSPR) here: http://xxxxxxxxxxxxxxxxx/warboard/anotebk.asp?proj=1786&Type=Active

This enhancement will modify various VistA applications, primarily VistA FB and IB. The scope of these enhancements involves an interaction between the various modules that would automate current manual processes where VA Medical Centers’ (VAMCs) staff are creating invoices to bill Veterans’ third party insurance co-payments for cost recoverable services provided for Non-VA health care services.

## User Profiles

User profiles are elaborated in Table 1:

Table : Primary and Secondary Users

| Name | Description | Responsibilities |
| --- | --- | --- |
| Primary Users | VAMC FB and Revenue/ Medical Care Cost Recovery (MCCR) staff | Process NVC authorizations and claims for the Veterans’ Non-VA healthcare services |
| Secondary Users | CBO National Fee Program Office (NFPO) and Purchased Care Program Office Staff, Veterans Integrated Service Networks (VISN) Staff and VA Central Office (VACO) CBO Staff | Monitor, track, trend, analyze and report data on the VA care |

# Background

## Overview of the System

The VA, VHA is one of the world’s largest health care delivery organizations. As part of an integrated strategy to provide Veterans with timely access to quality health care services, VA clinics are authorized to pay for health care services acquired from Non-VA health care providers. These services may be acquired by eligible Veterans from Non-VA health care providers when access to services is not available from VA. The VA is legislatively “mandated to appropriately reimburse for health care for eligible beneficiaries.”

The VA manages NVC authorization, claims processing and reimbursement for services acquired from Non-VA health care providers. The basic provisions of and authority for the NVC program are provided by public laws passed by Congress. These laws have been codified into the United States Code and further clarified in the Code of Federal Regulations. Various VA policies and directives have been written to provide additional interpretation.

In general, while Veterans seeking health care at VA expense should obtain such care at VA facilities, Non-VA facility care, or the use of FB, is generally only authorized when appropriate VA services and/or facilities are not available or cannot be economically provided to the Veteran due to geographical inaccessibility. The use of FB, or Non-VA facilities, is not considered a permanent solution for Veteran health care needs.

The VA Office of Inspector General (OIG) recently reported that VHA is not optimizing revenue intake via the Medical Care Collection Fund (MCCF) billing of NVC claims cases. The VHA CBO, Purchased Care Program Office is in agreement and affirms that the current VistA does not support the business processes for securing potential cost recovery of billable Non-VA episodes of care in a timely and efficient manner. They are seeking enhancements to the VistA FB and IB/Revenue software applications to automate the movement of billable Non-VA cases from VistA FB through the Revenue IB software product, to include prioritization of high cost claims, automation of Veterans’ Co-Pay requirements and decision support for assessing clinical review of cases.

## Overview of the Business Process

A high-level view of the current (as-is) and future (to-be) Fee Revenue Enhancement processes/workflows are available via TSPR here: x

## Overview of the Significant Requirements

The Fee-Revenue Enhancements BRD, Requirements Specification Document (RSD), and Requirements Traceability Matrix (RTM) are available via TSPR here: [x](http://tspr.vista.med.va.gov/warboard/anotebk.asp?proj=1786&Type=Active)

Until documents are approved and posted to TSPR, they will be made available upon request from the project team.

### Overview of Functional Requirements

The Fee-Revenue Enhancements project will execute the following:

* Enhancement provides an interaction from FB to IB to support the VHA Revenue Staff in Revenue Pre-Certification.
* This enhancement provides an interaction into FB to support the VHA Revenue Staff in the creation/revision of claims.

This document addresses only the *Fee Revenue* enhancement, one of five enhancements comprising the EDI CPAC Revenue Enhancements Program. Details regarding other enhancements in the program are described in separate RSDs.

During requirements elaboration for this enhancement, the Business Needs (BNs) and requirements were discussed from the perspective of the BRD, including the changes recommended in the BRD to address the BNs.

The revised understanding of the BNs and goals has revealed that BNs 2 FRE-2.10 – FRE-2.20 are no longer needed since they have been addressed on other projects and don’t need to be implemented. These revised BNs will be reflected in the RSD. **The RSD should be regarded as the source for the most recent BNs and requirements based on requirements elaboration.**

The BNs/requirements are shown in Table 2. Table 2 also reflects the obsolescence of BN2 requirements FRE-2.10 – FRE-2.20 that are no longer needed on this Enhancement.

Table : Functional Requirements

| BN | Owner Requirement (OWNR)  Number | OWNR | Priority\* |
| --- | --- | --- | --- |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.1 | The system shall have the ability to record the authorization and the corresponding authorization modification data. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.2 | The system shall have the ability to transmit the authorization and the corresponding authorization data to the IB package to support Revenue pre-certification. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.3 | The system shall have the ability to display the authorization and the corresponding authorization data in real-time to include any authorization updates. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.4 | The system shall have the ability to automatically alert/notify specified Revenue users of instances of authorized care for members with insurance. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.5 | The system shall have the ability to track all transmissions of authorizations and authorization updates to the IB package. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.6 | The system shall have the ability to transmit all claim data received and processed to the IB package. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.7 | The system shall have the ability to track all transmissions of claims and claims updates to the IB package. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.8 | The system shall have the ability to identify paid claims associated with a patient whose eligibility is copay required. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.9 | The system shall have the ability to *interact* with the VistA IB Software (bi-directional) to send NVC payment data on paid claims for first party (co-pay) and third party determination. | High |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.10 | The system shall have the ability to receive cost recovery and recoupment detail updates for NVC third party accounts receivables from the IB package. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.11 | The system shall have a bi-directional *interaction* with Financial Management System (FMS) to process recoupment data for NVC claims with overpayments, duplicate payments and erroneous payments. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.12 | The system shall have the ability to create workflow for all issues that must be resolved manually. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.13 | The system shall have the ability to manage all workflow items resulting from issues. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.14 | The system shall have the ability to track all workflow, including date and time stamps for each workflow stage throughout the workflow life cycle for historical and reporting purposes. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.15 | The system shall have the ability to escalate/reassign workflow items based on user specific permissions. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.16 | The system shall provide the capability to query workflow results both in detail and in summary. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.17 | The system shall have the ability for automatic and manual report generation. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.18 | The system shall have the ability for all reports to be archived/stored in a depository for historical purposes to be accessed on demand by any user with the appropriate permissions. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.19 | The system shall have the ability for designated users to create ad hoc reports. | Not Required |
| BN 2: The system shall have the ability to *interact* with the VistA IB Software to transmit authorized care data at the time the care is authorized and paid claims data, post NVC payment. | FRE-2.20 | The system shall have the ability to allow all users with the appropriate permissions to export data into other formats, such as Access data tables and/or Excel workbooks for analysis and manipulation. | Not Required |

### Overview of Functional Workload/Performance Requirements

Not applicable – this project is composed of VistA FB and IB patches. There are no functional workload/performance requirements associated with this project.

### Overview of Operational Requirements

Not applicable – there are no changes to existing operational requirements.

### Overview of Technical Requirements

Not applicable – there are no technical requirements that drive the conceptual design.

### Overview of Security or Privacy Requirements

All VA and VHA security or privacy requirements will be adhered to. Cross-cutting security requirements are contained in the VA Enterprise Requirements Repository (ERR). Efforts that involve the collection and maintenance of individually identifiable information must be covered by a Privacy Act system of records notice.

### Overview of System Criticality and High Availability Requirements

The FB and IB systems are composed of multiple VistA applications. Each of these applications has its own availability and downtime requirements which will be VAMC dependent. The FB and IB applications do not impose any requirement beyond the requirement already specified by each of the applications. Any related claim that cannot be submitted due to communication or other issues will be queued and submitted when the systems are available.

### Single Sign-on Requirements

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

### Requirement for Use of Enterprise Portals

Not applicable – there are no changes to the requirements for use of enterprise portals.

### Special Device Requirements

Not applicable – no special devices are required.

# Conceptual Design

This project is composed of VistA FB and IB patches and there are no conceptual design considerations. The actual software detailed design is presented in Section 6.2.2 Specific Requirements.

## Conceptual Application Design

Not applicable – there are no conceptual design considerations.

### Application Context

These enhancements do not require any changes to the application context.

Figure 1 provides the application context.

Fee Revenue Enhancement Application Context Diagram


Figure : Fee Revenue Enhancement Application Context Diagram

### High-Level Application Design

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

For informational purposes and application context, Figure 2 and Figure 3 display the Electronic Data Interchange (EDI) X12 messaging infrastructure for Veterans and beneficiaries.

Overview of EDI X12 Messaging

Figure : Overview of EDI X12 Messaging

*Overview of Claims Processing and Eligibility (CP&E) System and Interfaces*

Figure : Overview of Claims Processing and Eligibility (CP&E) System and Interfaces

### Application Locations

VistA is deployed universally across VHA at more than 1,500 sites of care, including each Veterans Affairs Medical Center (VAMC), Community Based Outpatient Clinic (CBOC) and Community Living Center (CLC), as well as at nearly 300 VA Vet Centers. The Fee Basis and Integrated Billing application modules are located within the nationally released VistA application. This will not change as a result of this project.

## Conceptual Data Design

### Project Conceptual Data Model

Figure 4 provides the conceptual data model for this project. The Billing Worklist is a new addition to the conceptual data model. Insurance Company billing, previously completed manually, will be automated as a result of this project.



Figure : Data Model

For a more detailed visual of the VistA data model, refer to the Entity Relationship Diagram (ERD) available via TSPR here:

### Database Information

The database being utilized for this project is VistA. Within the VistA database, two software modules will be utilized – Fee Basis and Integrated Billing.

Table : Database Information

|  |  |  |  |
| --- | --- | --- | --- |
| Database Name | Description | Type | Steward |
| VistA | Veterans Health Information Systems and Technology Architecture | I | VA |

The VistA Fee Basis module, and associated Massachusetts General Hospital Utility Multi-Programming System (Mumps) database, provides for more efficient and accurate operation of the fee for service program with reduction of paperwork, savings in staff hours, minimization of errors, and by allowing medical facilities to have greater control over disbursement of fee medical, pharmacy, and travel monies.

The VistA Integrated Billing module, and associated Mumps database, allows for the capture, maintenance, and storage of insurance data including policy information and related benefits. It provides the ability to electronically transmit bills to payers with the data required by HIPAA. It also provides the ability to receive and store electronic 835 Health Care Claim Payment/Advice. It includes the ability to create printed bills which can then be mailed to payers who are unable to accept an electronic claim. A Claims Tracking feature is available to assist utilization review staff in tracking episodes of care, completing pre-certifications, completing continued stay reviews, and processing appeals and denials. An Automated Biller module provides a process that automatically creates bills for billable events which can then be finished and authorized by the billing staff. The IB software also provides many reporting features that support the billing staff by providing statistics, tracking and historical information.

### User Interface Data Mapping

VistA Fee Basis receives data from the Fee Basis Claims System (FBCS), a Graphical User Interface (GUI) system that sends authorization and claim data to the VistA Fee Basis system. These fields are mapped in VistA, via Remote Procedure Calls (RPCs). There will be no changes to VistA Fee Basis interface mapping for this project.

The VistA Integrated Billing package electronically transmit bills to payers with the data required by HIPAA. It also provides the ability to receive and store electronic 835 Health Care Claim Payment/Advice. There will be no changes to VistA Integrated Billing interface mapping for this project.

## Conceptual Infrastructure Design

### System Criticality and High Availability

No changes are expected to system criticality and high availability as a result of this project. Reliability and availability requirements for VistA are managed by regional data centers.

### Special Technology

VistA has historically been built on a client-server architecture, which ties together workstations and personal computers with user interfaces at VA facilities. No special technologies will be modified as a result of this project.

### Technology Locations

All software changes for this project will be completed within nationally released VistA software code. Mumps routines (programs) will be modified and added to the Fee Basis and Integrated Billing software.

### Conceptual Infrastructure Diagram

#### Location of Environments and External Interfaces

Please see Figure 2 and Figure 3. Modifications for this project will be completed within the bounds of the VistA system, only.

#### Conceptual Production String Diagram

No conceptual production string diagram exists for the VistA Fee Basis and Integrated Billing modules. The standard VistA configuration will be utilized.

# System Architecture

This project is composed of VistA patches, which do not require any changes to the VistA System Architecture. However, brief descriptions of these entities will be provided below.

## Hardware Architecture

The FB and IB applications are part of the VistA suite of applications. The VistA Monograph states:

“VistA is deployed universally across VHA at more than 1,500 sites of care, including each VAMC, Community Based Outpatient Clinic (CBOC) and Community Living Center (CLC), as well as at nearly 300 VA Vet Centers. These locations will have a variety of hardware platforms but all meet a minimal standard required to support VistA.”

## Software Architecture

The FB and IB systems are composed of multiple VistA applications. They also interact with many other VistA standard modules, such as Mailman, Kernel, Health Level 7 (HL7), and FileMan. The entire VistA application consists of 200+ modules. The VistA monograph includes this description for VistA:

“VistA is an integrated Electronic Health Record (EHR) information technology system with application packages that share a common data store and common internal services. The data store and VistA kernel are implemented in the Massachusetts General Hospital Utility Multi-Programming System (MUMPS) computer language, and the Computerized Patient Record System (CPRS) GUI is implemented in Delphi. Application clients use a highly-efficient proprietary protocol to access data. VistA is highly configurable and customizable, and in addition to appropriate connectivity amongst VistA modules, VistA supports the integration of best-of-breed applications at multiple levels, including MUMPS Application Programming Interface (API), Remote Procedure Call (RPC), Medical Domain Web Services (MDWS), HL7, and data exchange via Blue Button or eHealth Exchanges. VistA comprises nearly 200 distinct applications/modules, 15,000 routines, and millions of lines of computer code.

The backbone of VHA’s clinical and administrative information technology capability, VistA has historically been built on a client-server architecture, which ties together workstations and personal computers with GUIs at VA facilities. The CPRS GUI is as highly customizable and runs on workstations, laptops, tablets (including iPads) and smart phones. VistA interoperates with numerous Commercial Off-the-Shelf (COTS) software applications and with selected information technology systems of other federal agencies and, increasingly, health information exchange networks. At the time of publication of this edition of the Monograph, comprehensive proposed enhancements to VistA were in the initial stages; referred to as “VistA Evolution” these enhancements are will reflect development and architecture enhancements to allow greater interaction with data and greater efficiency for the VistA system.”

Note that the changes required by this project are based in the MUMPS computer language and do not impact CPRS.

## Network Architecture

*Overview of Claims Processing and Eligibility (CP&E) System and Interfaces*

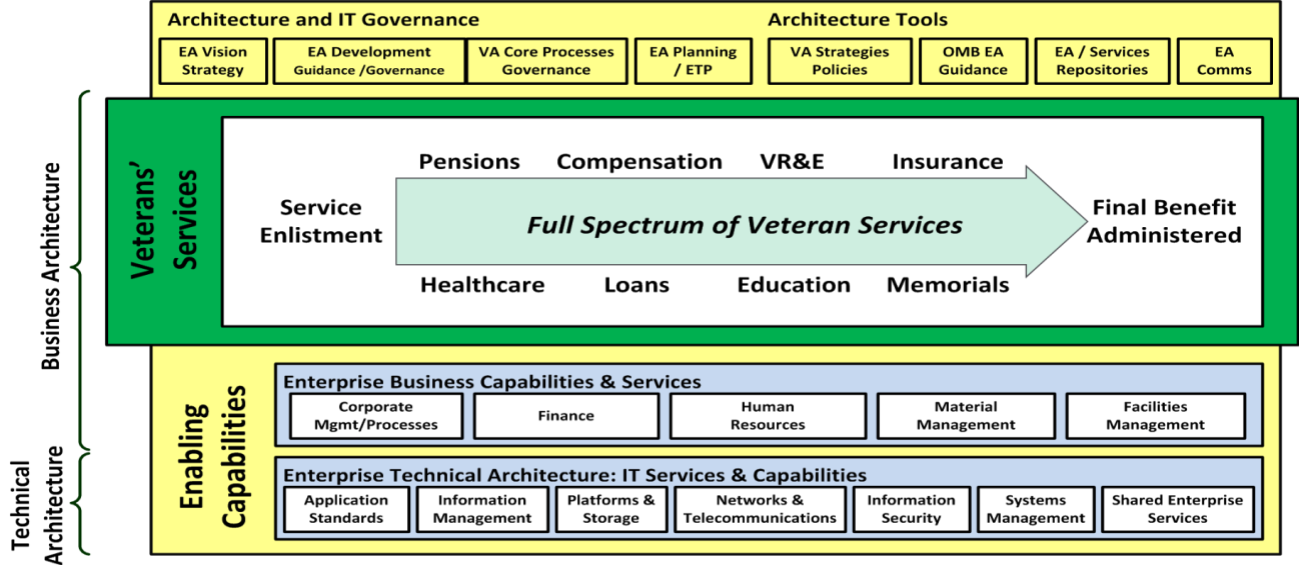
**Figure 5 – Network Architecture**

Please see Figure 5, above, for a depiction of network architecture surrounding Claims/Billing processing. The network architecture is not impacted or modified by this project.

## SOA/ESS

Not applicable – this project will not impact or mandate any additional requirement for Service Oriented Architecture (SOA)/Enterprise Shared Services (ESS).

## Enterprise Architecture



**Figure 6 – Enterprise Architecture**

Figure 6, above is a depiction of the VistA system’s Enterprise Architecture. No modifications are to me made to the existing system’s (VistA’s) architecture, as a result of this project.

# Data Design

## Database Management System (DBMS) Files

This is a VistA Integrated Billing patch. The database management system (DBMS) in use is VA FileMan.

The VistA Integrated Billing package uses the standard VA FileMan data dictionaries (DDs) and files. This table identifies the VA FileMan databases which are being modified by this VistA Integrated Billing project.

Table : FileMan Databases

| Database | File Number |
| --- | --- |
| FEE BASIS PATIENT FILE - AUTHORIZATION SUB-FILE (New Trigger) | 161.01 |
| IB-FB INTERFACE TRACKING FILE (New File)  *Note: This file allows IB and FB to interact with each other.* | 360 |

* + 1. **FEE BASIS PATIENT FILE – AUTHORIZATION SUB-FILE (#161.01) – New Trigger**

A new trigger / cross-reference (Field: FROM DATE, 161.01,.01) has been added so that each time a new or modified authorization is set in VistA, a new record will log to the IB-FB INTERFACE TRACKING FILE (#360).

Note: Trigger code (EVENT^IBFBUTIL) is detailed in the Routine Table portion of this document

STANDARD DATA DICTIONARY #161.01 -- AUTHORIZATION SUB-FILE

STORED IN ^FBAAA(D0,1,

DATA NAME GLOBAL DATA

ELEMENT TITLE LOCATION TYPE

-------------------------------------------------------------------------------

161.01,.01 FROM DATE 0;1 DATE

INPUT TRANSFORM: S %DT="EX" D ^%DT S X=Y K:Y<1 X

LAST EDITED: OCT 05, 2015

DESCRIPTION: The beginning date for which the patient is

authorized to obtain services from private

vendors. For every claim made by a vendor for

this patient, this date will be checked to

insure that the date the service was provided

was on or after this date.

DELETE TEST: .01,0)= I $$DELA^FBUCDD1(.DA,1)

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

CROSS-REFERENCE: 161^AIC1^MUMPS

1)= S:$P(^FBAAA(DA(1),1,DA,0),U,20)]"" ^FBAAA("

AIC",DA(1),-X,$P(^FBAAA(DA(1),1,DA,0),U,20),DA)

=""

2)= K:$P(^FBAAA(DA(1),1,DA,0),U,20)]"" ^FBAAA("

AIC",DA(1),-X,$P(^FBAAA(DA(1),1,DA,0),U,20),DA)

Cross-reference used to identify potential

billables using inverse date. Used to set

"AIC" cross-reference on field .097, POTENTIAL

COST RECOVERY CASE.

CROSS-REFERENCE: 161^ATST

1)= S ^FBAAA("ATST",$E(X,1,30),DA(1),DA)=""

2)= K ^FBAAA("ATST",$E(X,1,30),DA(1),DA)

CROSS-REFERENCE: 161.01^B

1)= S ^FBAAA(DA(1),1,"B",$E(X,1,30),DA)=""

2)= K ^FBAAA(DA(1),1,"B",$E(X,1,30),DA)

CROSS-REFERENCE: 161^AENR01^MUMPS

1)= D:'$D(DIU(0)) EVENT^IVMPLOG(DA(1))

2)= D:'$D(DIU(0)) EVENT^IVMPLOG(DA(1))

Notify HEC of changes to FEE Authorization.

CROSS-REFERENCE: 161^AC^MUMPS

1)= D:'$D(DIU(0)) EVENT^IBFBUTIL(DA(1))

2)= D:'$D(DIU(0)) EVENT^IBFBUTIL(DA(1))

Add new entry into file 360 on new Authorization

RECORD INDEXES: AMRA (#1327), AUD (#1204)

* + 1. **IB-FB INTERFACE TRACKING FILE (#360) – New File**

STANDARD DATA DICTIONARY #360 -- IB-FB INTERFACE TRACKING FILE

STORED IN ^IBFB(360,

DATA NAME GLOBAL DATA

ELEMENT TITLE LOCATION TYPE

-----------------------------------------------------------------------------

DD ACCESS:

RD ACCESS:

WR ACCESS:

DEL ACCESS:

LAYGO ACCESS:

AUDIT ACCESS: @

IDENTIFIED BY: PATIENT (#.02), FB AUTHORIZATION (#.03)

NON-VA CARE field (#.33) of the CLAIMS TRACKING File (#356)

CROSS REFERENCED BY: AUTH DELETION DATE/TIME (AC), CLAIM NUMBER(AD), ENTRY NUMBER(B), PATIENT(C)

INDEXED BY: BI QUEUE (BI), READY TO BILL (BL), PATIENT & FB AUTHORIZATION

(D), INVOICE NUMBER & FB PAYMENT INDICATOR (F), FR QUEUE (FR),

IV QUEUE (IV), SC QUEUE (SC), UR QUEUE (UR)

360,.01 ENTRY NUMBER 0;1 NUMBER (Required)

INPUT TRANSFORM: S:$D(X) DINUM=X

LAST EDITED: DEC 15, 2015

HELP-PROMPT: Enter an entry number.

DESCRIPTION: This is the internal entry number to identify

the link between claims/bills and Fee Basis

payments and authorizations.

TECHNICAL DESCR: Manual cross references:

; ^IBFB(360,"DFN",DFN,DT,IEN,IBLOG)="" Log

cross reference by Patient

; ^IBFB(360,"DT",DT,DFN,IEN,IBLOG)="" Log

cross reference by Date

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

CROSS-REFERENCE: 360^B

1)= S ^IBFB(360,"B",$E(X,1,30),DA)=""

2)= K ^IBFB(360,"B",$E(X,1,30),DA)

360,.02 PATIENT 0;2 POINTER TO PATIENT FILE (#2)

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter a Patient name.

DESCRIPTION: Patient name to link authorizations, payments

and claims/bills.

CROSS-REFERENCE: 360^C

1)= S ^IBFB(360,"C",$E(X,1,30),DA)=""

2)= K ^IBFB(360,"C",$E(X,1,30),DA)

RECORD INDEXES: D (#1334)

360,.03 FB AUTHORIZATION 0;3 NUMBER

INPUT TRANSFORM: K:+X'=X!(X>99999999)!(X<1)!(X?.E1"."1.N) X

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter a Patient Authorization number.

DESCRIPTION: Enter an Authorization IEN to link

authorizations, payments and claims/bills.

TECHNICAL DESCR: This is the Authorization number not a pointer.

It references file 161.01

RECORD INDEXES: D (#1334)

360,.04 AUTH DELETION DATE/TIME 0;4 DATE

INPUT TRANSFORM: S %DT="ET" D ^%DT S X=Y K:X<1 X

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter the date/time the Authorization was

deleted.

DESCRIPTION: This is the date that an Authorization was

deleted completely. Used for tracking purposes.

TECHNICAL DESCR: We're tracking in an Authorization was deleted

and there were pointers to claims or bills

present.

CROSS-REFERENCE: 360^AC

1)= S ^IBFB(360,"AC",$E(X,1,30),DA)=""

2)= K ^IBFB(360,"AC",$E(X,1,30),DA)

Payment ID cross reference used for linking FB

to IB.

360,.05 FB PAYMENT INDICATOR 0;5 FREE TEXT

INPUT TRANSFORM: K:$L(X)>25!($L(X)<1) X

LAST EDITED: DEC 04, 2015

HELP-PROMPT: Enter a payment indicator

DESCRIPTION: Fee Basis Payment indicator, formatted as

following: Patient IEN;Vendor IEN;Initial

treatment date IEN;Service provided IEN

Example: " 100000;455;2;4"

TECHNICAL DESCR: This represents a complex pointer to the FB

payment file (#162). Where the ID could be

saved as (for example) "165;222;1;10". Patient

pointer is 165, vendor pointer is 222, Service

date is 1, service provided is 10. Actual

pointer would be: ^FBAAC(165,1,222,1,1,1,10).

RECORD INDEXES: F (#1335)

360,1.01 CLAIM NUMBER 1;1 POINTER TO CLAIMS TRACKING FILE (#356)

LAST EDITED: DEC 16, 2015

HELP-PROMPT: Enter a claim ID.

DESCRIPTION: Enter a Claim ID to assist in linking fee basis

authorizations and payments to claims in file

#356.

TECHNICAL DESCR: We need a claim ID to assist in linking fee

basis authorizations and third party claims to

payments. Without this field we lose the

ability to link 3rd party claims to fee basis.

CROSS-REFERENCE: 360^AD

1)= S ^IBFB(360,"AD",$E(X,1,30),DA)=""

2)= K ^IBFB(360,"AD",$E(X,1,30),DA)

360,1.02 BILL/CLAIMS 1;2 POINTER TO BILL/CLAIMS FILE (#399)

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter an existing Bill for the Patient.

DESCRIPTION: We use this to link an existing entry in file

#399 to an existing fee basis authorization and

payment.

360,1.03 INVOICE NUMBER 1;3 POINTER TO FEE BASIS INVOICE FILE (#162.5)

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter a fee basis invoice number.

DESCRIPTION: We need to be able to link a fee basis invoice

to a claim/bill and to an existing

authorization and payment.

TECHNICAL DESCR: We need a Invoice ID to assist in linking fee

basis authorizations and claims/bills to

payments.

RECORD INDEXES: F (#1335)

360,1.04 INTEGRATED BILLING ACTION 1;4 POINTER TO INTEGRATED BILLING ACTION FILE (#350)

LAST EDITED: DEC 03, 2015

HELP-PROMPT: Enter a copayment associated with the FB auth

DESCRIPTION: Enter a first party copay IEN to link to a FB

Auth and payment.

TECHNICAL DESCR: We need a Bill ID to assist in linking fee

basis authorizations and first party claims to

payments. Without this field we lose the

ability to link 1st party claims to fee basis

Authorizations and payments.

360,2.01 IV QUEUE 2;1 SET

Insurance Verification Precert Queue

'IV' FOR IV QUEUE;

'XX' FOR REMOVED;

LAST EDITED: DEC 15, 2015

HELP-PROMPT: IV Precert Queue

DESCRIPTION: This is the Insurance Verification

Precertification Queue

FIELD INDEX: IV (#1337) REGULAR IR LOOKUP & SORTING

Short Descr: IV QUEUE

Set Logic: S ^IBFB(360,"IV",$E(X,1,240),DA)=""

Kill Logic: K ^IBFB(360,"IV",$E(X,1,240),DA)

Whole Kill: K ^IBFB(360,"IV")

X(1): IV QUEUE (360,2.01) (Subscr 1) (Len 240)

(forwards)

360,2.02 UR QUEUE 2;2 SET

RUR Precert Queue

'UR' FOR UR QUEUE;

'XX' FOR REMOVED;

LAST EDITED: DEC 15, 2015

HELP-PROMPT: UR Precert Queue

DESCRIPTION: This is the RUR Precertification Queue

FIELD INDEX: UR (#1339) REGULAR IR LOOKUP & SORTING

Short Descr: UR QUEUE

Set Logic: S ^IBFB(360,"UR",$E(X,1,240),DA)=""

Kill Logic: K ^IBFB(360,"UR",$E(X,1,240),DA)

Whole Kill: K ^IBFB(360,"UR")

X(1): UR QUEUE (360,2.02) (Subscr 1) (Len 240)

(forwards)

360,2.03 FR QUEUE 2;3 SET

Facility Revenue Billing Queue

'FR' FOR FR QUEUE;

'XX' FOR REMOVED;

LAST EDITED: DEC 15, 2015

HELP-PROMPT: FR Billing Queue

DESCRIPTION: This is the Facility Revenue Billing Queue

FIELD INDEX: FR (#1336) REGULAR IR LOOKUP & SORTING

Short Descr: FR QUEUE

Set Logic: S ^IBFB(360,"FR",$E(X,1,240),DA)=""

Kill Logic: K ^IBFB(360,"FR",$E(X,1,240),DA)

Whole Kill: K ^IBFB(360,"FR")

X(1): FR QUEUE (360,2.03) (Subscr 1) (Len 240)

(forwards)

360,2.04 SC QUEUE 2;4 SET

RUR Service Connected Billing Queue

'SC' FOR SC QUEUE;

'XX' FOR REMOVED;

LAST EDITED: DEC 15, 2015

HELP-PROMPT: SC Billing Queue

DESCRIPTION: This is the RUR Service Connected Billing Queue

FIELD INDEX: SC (#1338) REGULAR IR LOOKUP & SORTING

Short Descr: SC QUEUE

Set Logic: S ^IBFB(360,"SC",$E(X,1,240),DA)=""

Kill Logic: K ^IBFB(360,"SC",$E(X,1,240),DA)

Whole Kill: K ^IBFB(360,"SC")

X(1): SC QUEUE (360,2.04) (Subscr 1) (Len 240)

(forwards)

360,2.05 BI QUEUE 2;5 SET

Billing Review Queue

'BI' FOR BI QUEUE;

'XX' FOR REMOVED;

LAST EDITED: DEC 15, 2015

HELP-PROMPT: BI Billing Queue

DESCRIPTION: This is the Billing Review Queue

FIELD INDEX: BI (#1332) REGULAR IR LOOKUP & SORTING

Short Descr: BI QUEUE

Set Logic: S ^IBFB(360,"BI",$E(X,1,240),DA)=""

Kill Logic: K ^IBFB(360,"BI",$E(X,1,240),DA)

Whole Kill: K ^IBFB(360,"BI")

X(1): BI QUEUE (360,2.05) (Subscr 1) (Len 240)

(forwards)

360,3.01 NEXT REVIEW DT 3;1 DATE

Date of next RUR precertification review

INPUT TRANSFORM: S %DT="E" D ^%DT S X=Y K:X<1 X

LAST EDITED: DEC 15, 2015

HELP-PROMPT: (No range limit on date)

DESCRIPTION: This field holds the date of the next RUR

precertification review. If a date has been

entered in this field, the authorization will

not display on the RUR precertification work

queue until the date specified.

360,3.02 1ST OR 3RD PTY 3;2 NUMBER

First Party Copay or Third Party Insurance

INPUT TRANSFORM: K:+X'=X!(X>3)!(X<1)!(X?.E1"."1.N) X

LAST EDITED: DEC 15, 2015

HELP-PROMPT: First party copay or third party insurance

DESCRIPTION: This field indicates if the bill is for first

party copay or third party insurance.

TECHNICAL DESCR: First Party Copay =

$P(^IBFB(360,IEN,3),"^",2)=1 Third Party

Insurance = $P(^IBFB(360,IEN,3),"^",2)=3

360,4 LOG 4;0 DATE Multiple #360.04

(Add New Entry without Asking)

360.04,.01 DATE/TIME 0;1 DATE

INPUT TRANSFORM: S %DT="ESTR" D ^%DT S X=Y K:X<1 X

LAST EDITED: NOV 05, 2015

HELP-PROMPT: Enter the date/time of the change to the

entries.

DESCRIPTION: Used to capture the changes to the entries in

the file, by date and time of change.

360.04,.02 EVENT 0;2 FREE TEXT

INPUT TRANSFORM: K:$L(X)>45!($L(X)<1) X

LAST EDITED: NOV 05, 2015

HELP-PROMPT: Enter the event that occurred that we logged.

DESCRIPTION: Used to track what kind of event occurred

that we are tracking.

360.04,.03 USER 0;3 POINTER TO NEW PERSON FILE (#200)

LAST EDITED: NOV 05, 2015

HELP-PROMPT: Enter the user that made the change.

FILES POINTED TO FIELDS

BILL/CLAIMS (#399) BILL/CLAIMS (#1.02)

CLAIMS TRACKING (#356) CLAIM NUMBER (#1.01)

FEE BASIS INVOICE (#162.5) INVOICE NUMBER (#1.03)

INTEGRATED BILLING ACTION (#350) INTEGRATED BILLING ACTION (#1.04)

NEW PERSON (#200) LOG:USER (#.03)

PATIENT (#2) PATIENT (#.02)

INDEX AND CROSS-REFERENCE LIST -- FILE #360

File #360

Record Indexes:

D (#1334) RECORD REGULAR IR LOOKUP & SORTING

Short Descr: Lookup index

Set Logic: S ^IBFB(360,"D",X(1),X(2),DA)=""

Kill Logic: K ^IBFB(360,"D",X(1),X(2),DA)

Whole Kill: K ^IBFB(360,"D")

X(1): PATIENT (360,.02) (Subscr 1) (forwards)

X(2): FB AUTHORIZATION (360,.03) (Subscr 2) (forwards)

F (#1335) RECORD REGULAR IR LOOKUP & SORTING

Short Descr: Cross reference by invoice and payment ID

Set Logic: S ^IBFB(360,"F",X(1),$E(X(2),1,30),DA)=""

Kill Logic: K ^IBFB(360,"F",X(1),$E(X(2),1,30),DA)

Whole Kill: K ^IBFB(360,"F")

X(1): INVOICE NUMBER (360,1.03) (Subscr 1) (forwards)

X(2): FB PAYMENT INDICATOR (360,.05) (Subscr 2) (Len 30)

(forwards)

INPUT TEMPLATE(S):

PRINT TEMPLATE(S):

SORT TEMPLATE(S):

FORM(S)/BLOCK(S):

## Non-DBMS Files

N/A. There are no non-DBMS Files in use.

## Data View

This project is a VistA Integrated Billing patch. We have an Entity Relationship Diagram (ERD) as one of the deliverables with this project.

For a more detailed visual of the VistA data model, refer to the ERD available via TSPR here:

# Detailed Design

## Hardware Detailed Design

N/A. This project is a VistA Integrated Billing patch. There are no modifications to any existing hardware

## Software Detailed Design

### Conceptual Design

#### Product Perspective

The VistA Integrated Billing system is a software package that is being modified to interact and share data with the VistA Fee Basis system. Data from NVC authorizations will be tracked and made available to the Integrated Billing system.

##### User Interfaces

User Interface with all Integrated Billing application software is performed through existing and new VistA screens. VistA is a character-based application accessible through terminal emulator software resident on networked computers.

The Integrated Billing application is a character-based system using VA FileMan as its database manager and Cache/M as the programming language. This software application is part of the VistA suite of applications, which include additional clinical and financial applications. There are no GUIs.

##### Hardware Interfaces

Communications between VistA sites and HIPAA EDI Services pass through the Local Area Networks (LANs) and across the VA intranet. The Services communicate over the Austin Information Technology Center (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

* Kernel V. 8.0
* MailMan V. 8.0
* VA FileMan V. 22.0
* Fee Basis V 3.5
* Integrated Billing V. 2.0

##### Communications Interfaces

##### Memory Constraints

N/A. There are no memory constraints associated with this project.

##### Special Operations

N/A. There are no special operations associated with this project.

#### Product Features

Please see Section 1.1 for a list of Scope Inclusions, Section 2.3 for an overview of the significant functional requirements, and Section 6.2.2 for a list of all of the functional requirements.

#### User Characteristics

In general, the resources that shall work with Integrated Billing software need to be knowledgeable in the area of Integrated Billing and, to a lesser extent, Fee Basis.

#### Dependencies and Constraints

N/A. There are no dependencies and constraints with this project.

### Specific Requirements

#### Database Repository

This project is a VistA patch. The database repository is standard VistA database using FileMan files and fields. Any modifications to this database repository are detailed in Section 6.2.2.2.

#### System Features

The following sub-sections of this SDD match the specific functional requirements as presented in the RSD for this project.

This table attempts to demonstrate how functional requirements from the Fee-Revenue Enhancements RSD relate to the design:

Table : Functional Requirements (RSD) vs. Design Elements

| Requirement | RSD Functional Req Item # | Design Elements |
| --- | --- | --- |
| The system shall have the ability to record the authorization and the corresponding authorization modification data in a new VistA file that will allow the IB system to "point" to the authorization data in Fee Basis system at the time the authorization is created. | FRFEE-2.1 | New trigger in the FEE BASIS PATIENT FILE (#161) will log new FB auth pointer into IB-FB Interface Tracking file |
| The system shall provide the ability to record authorization and authorization modification data in the IB system that includes:  Ordering Physician Diagnosis/Procedure DOS Non-VA Location All Active Insurance Last Date Insurance Verified Type of Service (NVC) Service Connected/Special Auth Elig | FRFEE-2.1-01 – FRFEE-2.1-08 | New trigger in the FEE BASIS PATIENT FILE (#161) will log new FB auth pointer into IB-FB Interface Tracking file |
| The system shall have the ability to interface (interact with) authorization and the corresponding authorization data to the IB package to support Revenue pre-certification. | FRFEE-2.2 | Nightly Process #1 will identify paid invoices and log them in the new IB-FB Interface Tracking File   These new entries / paid invoices will display on the main screen of the Facility Revenue Queue – Billing Worklist |
| The system shall provide the ability to transmit authorization and authorization modification data to the IB system that includes: Ordering Physician  Diagnosis/Procedure  DOS  Non-VA Location  All Active Insurance  Last Date Insurance Verified  Type of Service (NVC)  Service Connected/Special Auth Elig | FRFEE-2.2.01 - 2.2-08 | Nightly Process #1 will identify paid invoices and log them in the new IB-FB Interface Tracking File / Facility Revenue Billing Worklist Queue  These new entries will display on the main screen of the Facility Revenue Queue – Billing Worklist |
| The system shall have the ability to display the authorization and the corresponding authorization data (Fee Basis Claims System [FBCS] data elements available in VistA Fee) in real-time to include any authorization updates in a new IB Worklist. | FRFEE-2.3 | Billing Worklist and related screens |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the name of the ordering physician (NPI #, Taxonomy Code, Last Name, First Name MI). | FRFEE-2.3-01 | Billing Worklist - Action EE – Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the Diagnosis/Procedure for services authorized in free text with unlimited characters. | FRFEE-2.3-02 | Billing Worklist - Action EE – Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the scheduled date of service which include date ranges (MM/DD/YY). | FRFEE-2.3-03 | Main Billing Worklist Screen (DOS Start) Billing Worklist - Worklist Action EE – Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the name of Non-VA location (if available). | FRFEE-2.3-04 | Billing Worklist - Action EE - Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the billable insurance information on file. | FRFEE-2.3-05 | Billing Worklist - Action EE - Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the last date of billable insurance verified (MM/DD/YY). | FRFEE-2.3-06 | Billing Worklist - Action EE – Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the type of service (Now referred to as NVC – Type of Non-VA Care). | FRFEE-2.3-07 | Main Billing Worklist Screen  Billing Worklist - Action EE – Expand |
| The system shall provide the ability to display authorization and authorization data in IB system via Worklist that includes the existing special authority and/or service connected conditions in free text (if applicable). | FRFEE-2.3-08 | Billing Worklist - Action EE - Expand |
| The system shall have the ability to automatically alert/notify specified Revenue users of instances of authorized care for members with billable insurance using a new IB Worklist that has on demand refresh capabilities. | FRFEE-2.4 | Billing Worklist - Action EE – Expand |
| The system shall have the ability to track each transmission of authorizations and authorization updates to the IB package using the IB Worklist. | FRFEE-2.5 | Billing Worklist - Action HI – History New Report Option: IB NVC BILLING WORKLIST REPORT |
| The system shall provide an interface so once payment has been approved/issued (output of VistA Fee Basis), it would create/revise a revision of the paid Non-VA care claim and push into IB. | FRFEE-2.6 | Billing Queue / Billing Worklist - Action WA (Complete) – Worklist Actions (places in auto-bill queue)  Nightly Process #2 |
| The system shall provide the ability to track delivery of paid Non-VA care claims and claims updates from VistA Fee Basis to the IB package. | FRFEE-2.7 | IB-FB Interface Tracking File, Nightly Process #1 |
| The system shall provide the ability to identify unpaid/paid Non-VA care claims with the appropriate copay eligibility status, such as member "co-pay required" in the new IB Worklist. | FRFEE-2.8 | Billing Worklist Action EE – Expand |
| The system shall provide the ability to interface (interact with) Fee Basis with the VistA IB Software (bi-directional) to send NVC payment data on paid Non-VA claims for first party (co-pay) and third party determination via the new IB worklist. | FRFEE-2.9 | IB-FB Interface Tracking File, Nightly Process #2 |
| The system shall provide the ability to enter timeframe parameters when generating the IB Worklist based on paid date and date of service. | FRFEE-2.10 | New Report Option: IB NVC BILLING WORKLIST REPORT |

#### Nightly Process #1

The existing Means Test Nightly Compilation Job (^IBAMTC) calls the Claims Tracker - Auto-Enroller (^IBTRKR) while processing. A new process (^IBFBNP1) will be added to the Claims Tracker – Auto-Enroll jobstream. This process will loop through newly paid NVC invoices and add them to the IB-FB Interface Tracking File (#360) / Facility Revenue Queue of the Billing Worklist. The invoices will be further categorized by first-party copay and third-party insurance. This new nightly process will also tie NVC Authorizations and NVC Invoice IENs together (in the IB-FB Interface Tracking File (#360)) for downstream processing.

#### Billing Worklist

This enhancement seeks to assist the VAMCs in administering timely billing to first-party copay and third party payers by notifying appropriate staff as soon as Non-VA Care invoices are paid.

The system shall notify Facility Revenue personnel, Revenue Utilization Review (RUR) Nurses (Service Connected and Special Authority Eligibility Reviewers) and Billing personnel via a worklist, when a Non-VA Care invoice is paid on the VistA NVC “side”.

IB Billing Worklist process will be as follows:

* Non-VA Care invoice is paid by VA.
* Nightly Process #1 runs and places invoice on the **Facility Revenue Queue of the Billing Worklist** (FR Queue).
* The invoices will be further separated into First Party and Third Party Sub-Queues for the remaining worklist tasks.
* Facility Revenue personnel will review the invoice, using Billing Worklist link to Claims Tracking to assist with their task of determining if the NVC invoice should be removed from the worklist or proceed with RUR-SC and/or Billing Worklist review.
* Facility Revenue personnel will deem their review of the invoice “Complete” or elect to “Remove” the item from the billing process, via Worklist Action.
* If “Remove” action is undertaken by Facility Revenue personnel, invoice is logged in the IB-FB Interface Tracking File (#360) historical logging section (to be used for reporting) and removed from the Facility Revenue Queue and the overall Billing Worklist.
* If “Complete” action is selected by Facility Revenue personnel, invoice is logged, removed from the Facility Revenue Queue and displayed on the **RUR – SC/SA Queue of the Billing Worklist** (SC Queue) if the Veteran has SC/SA eligibility or the Billing Queue of the Billing Worklist (BI Queue) if the Veteran does not have SC/SA eligibility.
* An RUR Nurse will review the invoice for Service Connected and Special Authority care/treatment via Worklist links to Claims Tracking.
* The RUR Nurse may deem that billing can proceed (no Service Connected or Special Authority Eligibility) by using Worklist Action “Complete”. If the RUR Nurse deems that care/treatment was Service Connected or Special Authority, they will elect to “Remove” the invoice from the RUR – SC/SA Queue and the overall Billing Worklist.
* If “Complete” option was selected by the RUR Nurse, the invoice is recorded in a log file (with an optional RUR status message), removed from the RUR-SC/SA Queue, and placed in the Billing Queue.
* If “Remove” option was selected by the RUR Nurse, the invoice is recorded in a log file (with optional RUR status message), removed from RUR-SC/SA queue and removed from the Billing Worklist.
* Billing personnel will review invoices on the **Billing Queue of the Billing Worklist** (BI Queue).
* Billing personnel will do a final review of any invoices that have progressed to their queue – They may opt to use Worklist links to Claims Tracking to assist with their task of determining whether or not the invoice should proceed with NVC billing.
* If Billing personnel deem that the invoice review is “Complete”, a bill will be created. The bill will be accessible from the “Enter/Edit Billing Information” VistA Option.
* If Billing personnel deem that they should “Remove” the invoice from the billing process, the invoice will be logged with an appropriate message and removed from the Billing Queue of the Billing Worklist.

Please Note: The Facility Revenue / RUR-SC/SA / Billing Worklist process is a method of tracking and handling workflow. It was designed to notify and assist the three departments (Facility Revenue, RUR-SC/SA and Billing) with their daily tasks. The queuing process will always progress from Facility Revenue to Billing, UNLESS an invoice is removed by one of the departments (deemed not billable). If the final review by Billing deems that the invoice should indeed be billed (to either first party copay or third party insurance), a bill will be created.

There is a Worklist History Action that will provide a means to view Worklist authorization actions, as well as date/time and user. A report was written to display past worklist activities from the worklist log. The VistA Option for the new report is: IB NVC BILLING WORKLIST REPORT

#### Design Element Tables

##### Routines (Entry Points)

Table : (Grouping): Routines

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBUTIL – IB FB Utilities *Note: This routine contains the “Trigger” code that logs new VistA authorizations into the IB-FB Interface Tracking File (#360), as well as some other IB FB utilities.* | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.1, FRFEE-2.1-01 through FRFEE-2.1-08 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | DEMOS^IBFBWL1 BLD^IBFBWL2 |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | FEE BASIS PATIENT FILE AUTHORIZATION SUB FILE (#161.01)  IB-FB INTERFACE TRACKING FIL E(#360) FEE BASIS PROGRAM FILE (#161.8) FEE BASIS PAYMENT FILE (#162)  FEE BASIS PHARMACY INVOICE FILE (#162.1)  FEE BASIS INVOICE FILE (#162.5) | | | | |
| **Related Protocols** | None | | | | |
| **Related Integration Control Registrations (ICRs)** | In process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBUTIL ;ALB/RED - API for EDI-CPAC (IB\*2.0\*554) ;10/01/15  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;OCT 15, 2015;Build 3  ;Per VA Directive 6402, this routine should not be modified.  ;  ; Integration Agreements  ; 10000 NOW^%DTC  ; 2053 UPDATE^DIE  ;  Q ; Routine cannot be called directly  ;  ;Special note: We are creating and using a few new global nodes in fil.  ; If they are defined in FM we lose control over them, and a re-index .  ; They only exist as temporary flags for our work list functionality.  ;  ; ^IBFB(360,"DFN",DFN,DT,IEN,IBLOG)="" Log cross reference by Patient  ; ^IBFB(360,"DT",DT,DFN,IEN,IBLOG)="" Log cross reference by Date  ;  EVENT(DFN) ; IB\*2.0\*554  ;Input: DFN  ;Output: none  N DIKIEN,AUTH,IENS,IBFBDT,FDA,IBIENS,IBEVENT,IBLOG,IBMOD,DELFLG,IENROON  S (IEN,DIKIEN,IENS)=0  S DELFLG=$G(D) ;Kill entry flag  I $G(DA)'="",DA'=DFN S AUTH=DA  I $G(AUTH)="",$G(D1)'="" S AUTH=D1  S DK=$G(DK)  I 'DK,$G(DIVAL)="" S:$G(DIVALUE)'="" DIVAL=DIVALUE ;(From Date verif)  Q:'$G(DFN)  Q:'$G(AUTH)  S IBFBDT=$$NOW^XLFDT() ;Used for date/time  ;Add entry into IBFB TRACKING file (#360)  S IBIENS="+1,",IENS=$P(^IBFB(360,0),U,3)+1  ;For deleted Auth's remove a few entries and set a delete date  I DELFLG D  . K FDA  . S DIKIEN=$O(^IBFB(360,"D",DFN,AUTH,0)) Q:DIKIEN=""  . S FDA(360,DIKIEN\_",",.03)="@",FDA(360,DIKIEN\_",",.04)=IBFBDT ;If Aus  . K ^IBFB(360,"IV",DIKIEN)  ;Add/edit  I 'DELFLG D  . K FDA  . S FDA(360,IBIENS,.01)=IENS,FDA(360,IBIENS,.02)=DFN,FDA(360,IBIENS,.0T  . S IENROOT="" ; Adding new entry)  I 'DELFLG D UPDATE^DIE("","FDA","IENROOT")  I DELFLG D UPDATE^DIE("","FDA")  S IEN=+$G(IENROOT(1))  I 'IEN,$G(DIKIEN)'="" S IEN=$G(DIKIEN)  Q:'IEN |
| D ;SET LOG FILE ENTRIES  . K FDA N IENROOT S IENROOT=""  . S FDA(360.04,"+1,"\_IEN\_",",.01)=IBFBDT,FDA(360.04,"+1,"\_IEN\_",",.03)Z  . S IBMOD=0,IBLOG=$P($G(^IBFB(360,IENS,4,0)),U,3)  . I IBLOG'="" S IBMOD=1  . S IBEVENT=$S(DELFLG:"Auth deleted",IBMOD=0:"Auth log-IV queue",1:"Au)  . S FDA(360.04,"+1,"\_IEN\_",",.02)=IBEVENT  . D UPDATE^DIE("","FDA","IENROOT")  ;LOG ENTRY AND CROSS REFERENCES  I 'DELFLG S IEN=IENROOT(1) D  . ;Set IEN in IV field/cross-reference  . K FDA  . S FDA(360,IEN\_",",2.01)="IV"  . D UPDATE^DIE("","FDA")  ; These cannot easily be set in FM, we don't have a date and we can't N  S LOGIEN=0,LOGIEN=$P(^IBFB(360,IEN,4,0),U,3)  S ^IBFB(360,"DFN",DFN,IBFBDT,IEN,LOGIEN)="",^IBFB(360,"DT",IBFBDT,DFN,"  Q  ;  GETAUTH(IENS,AUTHARR) ; API to call Authorization Data  D GETS^DIQ(161.01,IENS,".01;.02;.021;.03;.04;.055;.06;.065;.07;.08;.08)  Q  ;  GETST(IEN) ; Get Start Date using Invoice  N IBFLDS,IBINIEN,IBINLN1,IBFPNO1  S IBINV=$$GET1^DIQ(360,IEN\_",",1.03,"I") ; Invoice #  S IBFPNO=$$GET1^DIQ(161.01,IBIEN\_",",.03,"I") ; NVC IEN (Type) on FEE)  ; For Billing Worklist Only, NVC may have changed -- Check Fee Basis P)  I IBINV'="" D  . S IBINIEN=$O(^FBAAC("C",IBINV,DFN,""))  . S IBINLN1=$O(^FBAAC("C",IBINV,DFN,IBINIEN,""))  . S IBINLN2=$O(^FBAAC("C",IBINV,DFN,IBINIEN,IBINLN1,""))  . S IBFPNO1=$$GET1^DIQ(162.03,IBINLN2\_","\_IBINLN1\_","\_IBINIEN\_","\_DFN\_)  . S IBFPNOT=$$GET1^DIQ(162.03,IBINLN2\_","\_IBINLN1\_","\_IBINIEN\_","\_DFN\_)  . I $G(IBFPNOT)'="" S IBFP=IBFPNOT  . S IBST=$$GET1^DIQ(162.02,IBINLN1\_","\_IBINIEN\_","\_DFN\_",",".01","I") e  . I IBST="",IBFPNO=3 D  .. S IBRX=0  .. F S IBRX=$O(^FBAA(162.1,IBINV,"RX",IBRX)) Q:IBRX="" D  ... S IBST=$$GET1^DIQ(162.11,IBRX\_","\_IBINV,2,"I") ; Fill Date  Q  ; |
| GETPAY(IEN) ; Get NVC Payment Data using Invoice  N IBFLDS,IBINIEN,IBINLN1,IBINLN2,IBFPNO1  S IBINV=$$GET1^DIQ(360,IEN\_",",1.03,"I") ; Invoice #  S IBFPNO=$$GET1^DIQ(161.01,IBIEN\_",",.03,"I") ; NVC IEN (Type) on FEE)  ; For Billing Worklist Only, NVC may have changed -- Check Fee Basis P)  I IBINV'="" D  . S IBINIEN=""  . S IBINIEN=$O(^FBAAC("C",IBINV,DFN,IBINIEN)) Q:IBINIEN="" D  .. S IBINLN1=""  .. S IBINLN1=$O(^FBAAC("C",IBINV,DFN,IBINIEN,IBINLN1)) Q:IBINLN1="" D  ... S IBINLN2=""  ... S IBINLN2=$O(^FBAAC("C",IBINV,DFN,IBINIEN,IBINLN1,IBINLN2)) Q:IBIND  .... S IBFPNO1=$$GET1^DIQ(162.03,IBINLN2\_","\_IBINLN1\_","\_IBINIEN\_","\_D)  .... I $G(IBFPNO1)'="" S IBFPNO=IBFPNO1  .... S IBFLDS="2;63;64;65"  .... D GETS^DIQ(162.03,IBINLN2\_","\_IBINLN1\_","\_IBINIEN\_","\_DFN\_",",IBF  Q  ; |
| CHKBILL(IBIN) ;Check for prior bill  N IBINV,IBFBDT,IBCLM,IBFBAU  S IBINV=$TR(IBIN," ","")  S IBFBDT=""  F S IBFBDT=$O(^IBFB(360,"DFN",DFN,IBFBDT)) Q:IBFBDT="" D  . S IBFBAU=""  . F S IBFBAU=$O(^IBFB(360,"DFN",DFN,IBFBDT,IBFBAU)) Q:IBFBAU="" D  .. I $P($G(^IBFB(360,IBFBAU,1)),U,3)=IBINV D  ... S IBCLM=$$GET1^DIQ(360,IBFBAU\_",",1.01,"I")  ... I IBCLM'="" S FBINAU=$$PRECRT^IBTRC1(IBCLM,18)  ... S FBBILL=$$GET1^DIQ(360,IBFBAU\_",",1.02,"I")  ... I FBBILL'="" S FBSKIP=1  Q  ;  GETDTS(IBIEN) ;Get Begin and End Dates from Authorization  S IBST=$$GET1^DIQ(161.01,IBIEN\_",",.01,"I")  S IBEND=$$GET1^DIQ(161.01,IBIEN\_",",.02,"I")  Q  ; |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBTRKR – Claims Tracker – Auto-Enroller | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.2, FRFEE-2.2.01 - 2.2-08, FRFEE-2.7 | | | |
| **Related Options** | IB MT NIGHT COMP | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^IBAMTC (Means Test Nightly Compilation Job) | D UPDATE^IBTRKR1 ; update claims tracking site parameters (random sampler)  D ^IBTRKR2 ; add scheduled admissions to tracking  D ^IBTRKR3 ; add rx refill to outpatient encounters  D ^IBTRKR4 ; add outpatient encounters to tracking  D ^IBTRKR5 ; add outpatient prosthetics item to tracking  D NRUN^IBFBNP1 ; add nightly process for Non VA Care (IB\*2.0\*554) |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | Not applicable to this specific modification. Call for NVC processing added to end of nightly process. | | | | |
| **Related Protocols** | None | | | | |
| **Related Integration Control Registrations (ICRs)** | No additional ICRs for this modification | | | | |
| **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 |
| --- |
| See Below |

|  |
| --- |
| *New Code:*  +163 D NRUN^IBFBNP1 ; add nightly process for Non VA Care(IB\*2.0\*554) |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBNP1 – FB to IB Interaction Nightly Process #1  This process loops through paid invoices from NVC providers and  sets up the IB Billing Worklist – Facility Revenue Queue for first party and third party services. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.2, FRFEE-2.2.01 - 2.2-08, FRFEE-2.7 | | | |
| **Related Options** | IB MT NIGHT COMP | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^IBTRKR | MKARRLTC^FBPCR4  SITE^FBAACO  STA^XUAF4  CATC^FBPCR  INSURED^FBPCR4  FPINPT^IBEFURF  FPOPV^IBEFURF  UPDATE^DIE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | CLAIMS TRACKING FILE (#356) IB-FB INTERFACE TRACKING FILE (#360) INTEGRATED BILLING ACTION FILE (#350)  IB SITE PARAMETERS (#350.9)  VA FORM 10-7078 FILE (#162.4)  FEE BASIS PAYMENT FILE (#162)  CLAIMS TRACKING TYPE (#356.6) | | | | |
| **Related Protocols** | None | | | | |
| **Related Integration Control Registrations (ICRs)** | In process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBNP1 ;ALB/RED- EDI-CPAC build 1st and 3rd party copayments ;10/01/15  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;OCT 15, 2015;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;  ; Integration Agreements  ; 10000 NOW^%DTC  ; 2053 UPDATE^DIE  ; 2171 - $$STA^XUAF4  ; $$CATC^FBPCR  ; MKARRLTC^FBPCR4  ; FPINPT^IBEFURF  ; FPOPV^IBEFURF  ; $$INSURED^IBCNS1  ;  Q  NRUN(IBSTDT) ; main entry point - nightly run job to look at existing FB paymen0  ;Start by looking for recent FB payments  ; Input is start date (FM format), defaults to today-1  ;  N IBVEN,IBSERVDT,IBCATC,IBSITE,IBMTC,FBARRLTC,IBDTPD,IBDUZ,IBREC,IBRECL  N IBBILL,IBCLAIM,IBFBDT,IBIENS,IBLOG,IENROOT,IENS,IBSERV,FRSTPRT,IBFBIY  S IBSTDT=$G(IBSTDT)\1  I IBSTDT'?7N S IBSTDT=""  S FBARRLTC="" D MKARRLTC^FBPCR4 ;build array needed later for POV in y  S (IBCATC,IBMTC,FRSTPRT)=0 ;(CAT C FLAG and IBMTC = Determine if patie  ;  D SITE^FBAACO S IBSITE=$P(FBSITE(1),U,3)  ;S IBN=$$PT^IBEFUNC(IBDFN) D UTIL^IBCA3,UTIL^IBOA32 ; check for allJ  I IBSTDT="" S IBSTDT=$P(^IBE(350.9,1,0),U,6)\1 ; set to last tn  F S IBSTDT=$O(^FBAAC("AK",IBSTDT)) Q:'IBSTDT D PAYMT  Q |
| PAYMT ;  N IBAUTH,IBPOV,IBINV  S (DFN,IBPOV,IBINV,IBDOS,IBCARETY)=0 F S DFN=$O(^FBAAC("AK",IBSTDT,DFD  . S IBVEN=0 F S IBVEN=$O(^FBAAC("AK",IBSTDT,DFN,IBVEN)) Q:'IBVEN D  .. S IBSERVDT=0 F S IBSERVDT=$O(^FBAAC("AK",IBSTDT,DFN,IBVEN,IBSERVDTD  ... S IBSERV=0 F S IBSERV=$O(^FBAAC("AK",IBSTDT,DFN,IBVEN,IBSERVDT,IBD  .... ; Set the temporary payment array to service date and the zero noV  .... Q:$G(^FBAAC(DFN,1,IBVEN,1,IBSERVDT,1,IBSERV,0))="" ; qud  .... Q:$G(^FBAAC(DFN,1,IBVEN,1,IBSERVDT,1,IBSERV,"FBREJ"))]"" ;paymend  .... S IBDTPD=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DFN\_)  .... Q:'IBDTPD ; quit if the date paid is NULL/Empty  .... S IBDOS=$$GET1^DIQ(162.02,IBSERVDT\_","\_IBVEN\_","\_DFN\_",",.01,"I")  .... S IBAUTH=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DFN\_4  .... S IBTYP=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DFN\_"e  .... S IBINV=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DFN\_")  .... Q:'IBINV ;Quit if there is no Invoice for this record  .... S IBCARETY=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DF8  .... I "^2^3^6^7^"'[IBCARETY Q ;Fee Program categories  .... S IBPOV=$$GET1^DIQ(162.03,IBSERV\_","\_IBSERVDT\_","\_IBVEN\_","\_DFN\_")  .... Q:'IBPOV  .... S IBCATC=$$CATC^FBPCR(DFN,IBDOS,IBPOV) ;determine 3rd party copas  .... S IBFBINS=$$INSURED^IBCNS1(DFN,IBDOS) ; Check for active insurane  .... D ADMIT,CHKOTPT,FILE  Q  ; |
| ADMIT ; check for inpatient 1st party bills  Q:$G(IBADMDT)=""  S IBBILL=0  K ^TMP("IBRBF",$J) ; kill of temp global before call  D FPINPT^IBEFURF(DFN,IBADMDT)  S IBBILL=$O(^TMP("IBRBF",$J,"FP",0))  K ^TMP("IBRBF",$J)  Q  ;  ;  FILE ; check payment against file #360  ;  N FDA,IBEDIT,IBRECZ,IBOAUTH,IBOCLM,IBOBILL,IBFLAG  S (IBOAUTH,IBEDIT,IBREC,IBFLAG,IENS,IBCLAIM)=0,IBBILL=$G(IBBILL)  I $G(IBAUTH)="" S IBAUTH="0"  ; check to see if the patient has that invoice, id  I IBINV,$D(^IBFB(360,"F",DFN,IBINV)) S IBREC=$O(^IBFB(360,"F",DFN,IBINd  ;  ; check to see if there is a record existing withy  I 'IBAUTH,$D(^IBFB(360,"C",DFN)) D  . S IBRECZ=0 ; set a temporary record number to check against f  . F S IBRECZ=$O(^IBFB(360,"C",DFN,IBRECZ)) Q:IBRECZ="" D  .. S IBOCLM=$P($G(^IBFB(360,IBRECZ,1)),U) ;Claim IEN exists for this d  .. S IBOBILL=$P($G(^IBFB(360,IBRECZ,1)),U,4) ; Bill IEN exists for thd  .. I IBOCLM!IBOBILL S IBREC=IBRECZ,IBRECZ="a" Q ;Found a record to edt  .. Q  ;  I ('FRSTPRT&'IBFBINS)!('IBCATC&'IBFBINS) Q ; Quit if no insurance any  ;  ; edit an existing record  I IBREC D  . K FDA  . S IBCLAIM=+$$GET1^DIQ(360,IBREC\_",",1.01,"I")  . ; D STUB ;check/create stub record in file #356  . I $$GET1^DIQ(360,IBREC\_",",1.03,"I")="" S FDA(360,IBREC\_",",1.03)=$Gr  . I $$GET1^DIQ(360,IBREC\_",",.05,"I")="" S FDA(360,IBREC\_",",.05)=IBDOe  . I IBBILL,$$GET1^DIQ(360,IBREC\_",",1.04,"I")="" S FDA(360,IBREC\_",",1n  . I IBCLAIM,$$GET1^DIQ(360,IBREC\_",",1.01,"I")'=IBCLAIM S FDA(360,IBREr  . I $$GET1^DIQ(360,IBREC\_",",2.03,"I")="" S FDA(360,IBREC\_",",2.03)="Fe  . Q:'$D(FDA)  . S IBFLAG=1 ; flag used for log file  . D UPDATE^DIE("","FDA")  . I 'IBCATC S IBBILL=$$GET1^DIQ(360,IBREC\_",",1.04,"I") I IBBILL D SETT  .;  ;  ; |
| I 'IBREC D  . K FDA,IENROOT  . S IBFLAG=1 ; flag used for log file  . S IBIENS="+1,",IENS=$P(^IBFB(360,0),U,3)+1  . S FDA(360,IBIENS,.01)=IENS,FDA(360,IBIENS,.02)=$G(DFN),FDA(360,IBIEN)  . S FDA(360,IBIENS,1.03)=$G(IBINV),FDA(360,IBIENS,.05)=IBDOS ;Invoicee  . I IBBILL S FDA(360,IBIENS,1.04)=IBBILL ;1st Party Co-pay - Admission  . I OTPTBIL S FDA(360,IBIENS,1.04)=OTPTBIL ;1st Party Co-pay - outpt  . S FDA(360,IBIENS,2.03)="FR" ;Set facility revenue worklist queue  . S (IENROOT,IENROOT(1))="" ; adding new entry)  . D UPDATE^DIE("","FDA","IENROOT")  . I IENROOT(1)'="" S IBREC=IENROOT(1)  . ; D STUB K FDA  . S FDA(360,IBREC\_",",1.01)=IBCLAIM  . D UPDATE^DIE("","FDA")  I IBREC,'FRSTPRT D  . Q:$P($G(^IBFB(360,IBREC,3)),U,2)=3  . K FDA  . S IBFLAG=1 ; flag used for log file  . S FDA(360,IBREC\_",",3.02)=3  . D UPDATE^DIE("","FDA")  ;  I FRSTPRT D  . Q:$P($G(^IBFB(360,IBREC,3)),U,2)=1  . K FDA  . S IENROOT="",IBFLAG=1 ; flag used for log file  . S FDA(360,IBREC\_",",3.02)=1  . D UPDATE^DIE("","FDA","IENROOT")  ;  I IBBILL!(OTPTBIL) D ; set pointer for Inpt or outpt y  . I IBBILL Q:$P(^IB(IBBILL,0),U,23)=IBREC ; exists and valid  . I OTPTBIL Q:$P(^IB(OTPTBIL,0),U,23)=IBREC  . K FDA  . S IBFLAG=1 ; flag used for log file  . S FDA(350,IBBILL\_",",.23)=IBREC ; set Non-Va Care value to the p0  . D UPDATE^DIE("","FDA")  ; |
| LOG ; set log (audit) file entries  N FDA,IBEVENT,IBMOD,IBDUZ  Q:'IBFLAG ;No changes were made  D NOW^%DTC S IBFBDT=%  S IBMOD=0,IBDUZ=$G(DUZ) I $G(IBDUZ)="" S IBDUZ=".5" ; Set user to pos)  S FDA(360.04,"+1,"\_IBREC\_",",.01)=IBFBDT,FDA(360.04,"+1,"\_IBREC\_",",.0)  S IBMOD=0,IBLOG=$P($G(^IBFB(360,IBREC,4,0)),U,3)+1  I IBLOG="" S IBMOD=1  S IBEVENT=$S(IBMOD=0:"Auth log-FR queue",1:"Auth mod-FR queue"),FDA(36T  D UPDATE^DIE("","FDA")  Q  ;  STUB ; look for third party claim pointer in file #356  ;  K IENROOT  I IBCLAIM,$$GET1^DIQ(360,IBREC\_",",1.03,"I")'=IBINV S IBCLAIM=0 ;1 inm  I IBCLAIM,$D(^IBFB(360,"AD",IBCLAIM)),$O(^IBFB(360,"AD",IBCLAIM,0))'=Id  I 'IBCLAIM!($$GET1^DIQ(356,IBCLAIM\_",",.33,"I")="") D Q ; in)  . K FDC,ZIENS,ZIEN  . I 'IBCLAIM S ZIENS="+1,",ZIEN=$P(^IBT(356,0),U,3)+1,IENROOT="" D  .. S FDC(356,ZIENS,.01)=IBSITE\_ZIEN,FDC(356,ZIENS,.02)=DFN ;IEN and Pt  .. S FDC(356,ZIENS,.06)=IBDOS,FDC(356,ZIENS,.2)=1 ;Date of service ane  . ;Edit an existing claim with no pointer  . I IBCLAIM S ZIENS=IBCLAIM\_","  . S FDC(356,ZIENS,.33)=IBREC ;Link back to file #360 (IB-FB INTERFACE)  . I IBCARETY D  .. I IBCARETY=2 S FDC(356,ZIENS,.18)=6 Q ;Outpatient  .. I IBCARETY=3 S FDC(356,ZIENS,.18)=8 Q ;Pharmacy  .. I IBCARETY=6!(IBCARETY=7) S FDC(356,ZIENS,.18)=7 Q ;Inpatient  . I IBCLAIM D UPDATE^DIE("","FDC") ;edit  . I 'IBCLAIM D  .. D UPDATE^DIE("","FDC","IENROOT")  .. I IENROOT(1)'="" S IBCLAIM=IENROOT(1) ;New  Q  ; |
| SETOTPT ; Look for first party claim pointer in file #360 to an exist Bill IEN  N FDA  ;  Q:'$D(^IB(IBBILL,0)) ; invalid pointer to file #350 or t  Q:$P(^IB(IBBILL,0),U,23)=IBREC ; pointer is present and valid  S FDA(350,IBBILL\_",",.23)=IBREC ; set Non-Va Care value to the poi0  D UPDATE^DIE("","FDA")  Q  ;  CHKOTPT ; check for Outpatient 1st party bills  K ^TMP("IBRBF",$J) ; kill of temp global before call  S (FRSTPRT,OTPTBIL)=0  D FPOPV^IBEFURF(DFN,IBDOS) Q:'$D(^TMP("IBRBF",$J))  S OTPTBIL=$O(^TMP("IBRBF",$J,"FP",0)),FRSTPRT=1 ;set outpt 1st party g  K ^TMP("IBRBF",$J)  ;  ;END OF IBFBNP1 |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWL – Main Precert and Billing Worklist Routine (Note: Some sections of this routine are used by the Billing Worklist only.) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.2, FRFEE-2.3, FRFEE-2.4,FRFEE-2.3-03, FRFEE-2.3-07 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | , |
| --- | --- | --- |
|  | Called by VistA Option - IB BILLING WORKLIST  Subroutines Called by ListMan Actions -  EXPAND  LINKCT – Link to CT ACTIONS HISTORY | 1. BLDWL^IBFBWL1 (Billing Worklist only – NSR #20110303) 2. BLDWL^IBFBWL5 (Precert Worklist only – NSR #20081010)  3. Refresh Action calls GETAUT,BLDWL in ^IBFBWL1/^IBFBWL5 to rebuild lists |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | INSTITUTION FILE (#4)  PATIENT FILE (#2)  FEE BASIS PATIENT AUTHORIZATION SUBFILE (#161.01)  CLAIMS TRACKING FILE (#356) | | | | |
| **Related Protocols** | List  IB BILLING WORKLIST  Menu Protocol  IB BILLING WORKLIST MENU  Action Protocols  IB BILLING WORKLIST REFRESH - REFRESH^IBFBWL  IB BILLING WORKLIST EXPAND - EXPAND^IBFBWL  IB BILLING WORKLIST CLAIMS TRACKING – LINKCT^IBFBWL  IB BILLING WORKLIST ACTIONS – ACTIONS^IBFBWL  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes - in process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWL ;ALB/PAW-IB BILLING Worklist ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  EN ; -- Main entry point for IB BILLING WORKLIST 🡸This is the entry point for the billing worklist  N FILTERS,IBGRP,IBDIVS,IBWLTYP  S IBWLTYP="B"  I '$$FILTER(.FILTERS) Q  S IBGRP=$P($G(FILTERS(0)),U,1)  K XQORS,VALMEVL  D EN^VALM("IB BILLING WORKLIST")  Q  ;  EN2 ; -- Main entry point for IB NVC PRECERT WORKLIST 🡸This is the entry point for precert  N FILTERS,IBGRP,IBDIVS,IBWLTYP  S IBWLTYP="P"  I '$$FILTER(.FILTERS) Q  S IBGRP=$P($G(FILTERS(0)),U,1)  K XQORS,VALMEVL  D EN^VALM("IB NVC PRECERT WORKLIST")  Q  ;  INIT ; Initialize variables for IB BILLING WORKLIST 🡸Initialization for billing worklist  D KILLGLB  D GETAUT^IBFBWL1(IBGRP) 🡸Calls ^IBFBWL1 to build main billing worklist screen  I '$D(^TMP("IBFBWL",$J)) D Q  . W !!,\*7,"There are no new cost recoverable invoices on file."  . S DIR(0)="E"  . D ^DIR  . S VALMQUIT=1  . D EXIT  S IBWLTYP="B"  D BLDWL^IBFBWL1  Q  ; |
| INIT2 ; Initialize variables for IB NVC Precert Worklist  D KILLGLB  D GETAUT^IBFBWL5(IBGRP)  I '$D(^TMP("IBFBWL",$J)) D Q  . W !!,\*7,"There are no new authorizations on file."  . S DIR(0)="E"  . D ^DIR  . S VALMQUIT=1  . D EXIT  S IBWLTYP="P"  D BLDWL^IBFBWL5  Q  ;  HDR ; Set header for IB BILLING Worklist  N IBDIVS,IBFST,IBIEN,IBXX,IBY  D CHKFILT  I IBWLTYP="B" D  . S VALMHDR(1)=$S(IBY=1:"Facility Revenue Review",IBY=2:"RUR SC/SA",IBY=3:"Billing")  . S VALMHDR(1)=VALMHDR(1)\_" ("\_$S($P(FILTERS(0),U,4)=1:"First Party Copay",1:"Third Party")\_")"  I IBWLTYP="P" D  . S VALMHDR(1)=$S(IBY=1:"Insurance Verification",IBY=2:"RUR Pre-certification")  S VALMHDR(2)="Selected Division(s): "\_IBDIVS  Q  ;  HDR2 ; Set header for IB NVC Precert Worklist  N IBDIVS,IBFST,IBIEN,IBXX,IBY  D CHKFILT  S VALMHDR(1)=$S(IBY=1:"Insurance Verification",IBY=2:"RUR Pre-certification")  S VALMHDR(2)="Selected Division(s): "\_IBDIVS  Q  ; |
| FILTER(FILTERS) ; Set up filters  ; Sets an array of filters to determine which entries to include in display  ; Input: None  ; Output:  ; Returns: 0 if the user entered '^' or timed out, 1 otherwise  ; If Billing - FILTERS(0) = fee basis group (1=Facility Revenue, 2=RUR  SC/SA, 3=Billing)^ 0 (all) 1 (selected) institutions ^ 0 (all) 1 (selected) patients ^ 1(First Party Copay) 2 (Third Party)  ; If Precert - FILTERS(0) = fee basis group (1=insurance verification,  2=RUR)^ 0 (all) 1 (selected) institutions ^ 0 (all) 1 (selected) patients  ; FILTERS(1) = inst ien ^ inst ien ^ etc...  ; FILTERS(2) = pat ien ^ pat ien ^ etc...  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,IBXX,X,XX,Y  K FILTERS  ;  ; Billing Department  I IBWLTYP="B" D  . S DIR(0)="S",DIR("A")="Select (F)acility Revenue, (R)UR SC/SA or (B)lling"  . S DIR("?",1)="Enter 'F' for Facility Revenue, 'R' for RUR SC/SA"  . S DIR("?")="or 'B' to for Billing."  . S $P(DIR(0),U,2)="F:Facility Revenue;R:RUR SC/SA;B:Billing"  . W ! D ^DIR K DIR  . I $G(DIRUT) Q  . S X=$$UP^XLFSTR(X)  . S FILTERS(0)=$S(Y="F":1,Y="R":2,Y="B":3,1:0)  ;  ; Pre-certification Department  I IBWLTYP="P" D  . S DIR(0)="S",DIR("A")="Select (I)nsurance Verification or (R)UR Pre-certification"  . S DIR("?",1)="Enter 'I' for insurance verification authorizations."  . S DIR("?")="Enter 'R' for RUR authorizations."  . S $P(DIR(0),U,2)="I:Insurance Verification;R:RUR Pre-certification"  . W ! D ^DIR K DIR  . I $G(DIRUT) Q  . S X=$$UP^XLFSTR(X)  . S FILTERS(0)=$S(Y="I":1,Y="R":2,1:0)  ; |
| ; First Party Copay or Third Party  I IBWLTYP="B" D  . S DIR(0)="S",DIR("A")="Select (F)irst Party Copay or (T)hird Party Insurance"  . S $P(DIR(0),U,2)="F:First Party Copay;T:Third Party Insurance"  . S DIR("?",1)="Enter 'F' for First Party Copay."  . S DIR("?")="Enter 'T' for Third Party Insurance."  . S $P(DIR(0),U,2)="F:First Party Copay;T:Third Party Insurance"  . W ! D ^DIR K DIR  . I $G(DIRUT) Q  . S X=$$UP^XLFSTR(X)  . S $P(FILTERS(0),U,4)=$S(Y="F":1,Y="T":3,1:0)  ;  I $G(DIRUT) Q 0  ;  ; Site (Division) Filter  S DIR(0)="S",DIR("A")="Select(A)ll or (S)elected Division(s):",DIR("B")="All"  S DIR("?",1)="Enter 'A' for ALL Divisions."  S DIR("?")="Enter 'S' to view entries for selected Division(s)."  S $P(DIR(0),U,2)="A:All Divisions;S:Selected Divisions"  W ! D ^DIR K DIR  I $G(DIRUT) Q 0  S X=$$UP^XLFSTR(X)  S $P(FILTERS(0),U,2)=$S(Y="A":0,1:1)  ;  ; Set Division Filter  I $P(FILTERS(0),U,2)=1 D ASKDIV(.FILTERS)  ;  ; Patient Filter  S DIR(0)="S",DIR("A")="Select(A)ll or (S)elected Patient(s):",DIR("B")="All"  S DIR("?",1)="Enter 'A' to select ALL Patients."  S DIR("?")="Enter 'S' to view entries for selected Patients."  S $P(DIR(0),U,2)="A:All Patients;S:Selected Patients"  W ! D ^DIR K DIR  I $G(DIRUT) Q 0  S X=$$UP^XLFSTR(X)  S $P(FILTERS(0),U,3)=$S(Y="A":0,1:1)  ; Set Patient / Veteran filter  I $P(FILTERS(0),U,3)=1 D ASKPAT(.FILTERS)  D SHOWFILT(.FILTERS)  Q 1  ; |
| ASKDIV(FILTERS) ; Sets a list of Divisions to be displayed in the Billing Worklist  ; Input: FILTERS - Current Array of filter settings  ; Output: FILTERS - Updated Array of filter settings  N DIC,DIR,DIRUT,DIVS,DUOUT,FIRST,IBIENS,IBIENS2,IBXX,IEN,N,X,Y  S DIC=4,DIC(0)="AE",FIRST=1  F D Q:+IEN<1  . D ONEDIV(.DIC,.IEN,.FIRST) ; One Division prompt  . Q:+IEN<1  . S IBIENS($P(IEN,U,2))=$P(IEN,U,7)  . S IBIENS2($P(IEN,U,1))=$P(IEN,U,2)  I '$D(IBIENS) S FILTERS(1)="" Q  ;  ; Set the filter node responses in alphabetical order  S IBXX=""  F S IBXX=$O(IBIENS2(IBXX)) Q:IBXX="" D  . S FILTERS(1,IBXX)=""  Q  ;  ONEDIV(DIC,IEN,FIRST) ; Prompts the user for a Division  ; Input: DIC - Variable/Array of settings needed for ^DIC call  ; FIRST - Set to 1 initially and then 0 for subsequent calls  ; Output: FIRST - Set to 0  ; IEN - IEN of the selected Division  ; null of no selection was made  S DIC("A")=$S(FIRST:"Select a Division: ",1:"Select Another Division: ")  D ^DIC  S FIRST=0,IEN=Y  Q  ; |
| ASKPAT(FILTERS) ; Sets a list of patients  ; Input: FILTERS - Current Array of filter settings  ; Output: FILTERS - Updated Array of filter settings  N DIC,DIR,DIRUT,DIVS,DUOUT,FIRST,IBIENS,IBIENS2,IBN,IBXX,IEN,X,Y  S DIC=2,DIC(0)="AE",FIRST=1  F D Q:+IEN<1  . D ONEPAT(.DIC,.IEN,.FIRST) ; One patient  . Q:+IEN<1  . S IBIENS($P(IEN,U,2))=$P(IEN,U,1)  . S IBIENS2($P(IEN,U,1))=$P(IEN,U,2)  I '$D(IBIENS) S FILTERS(2)="" Q  ;  ; Set the filter node responses in alphabetical order  S IBXX=""  F D Q:IBXX=""  . S IBXX=$O(IBIENS(IBXX))  . Q:IBXX=""  . S IBN=IBIENS(IBXX)  . S FILTERS(2,IBN)=""  Q  ;  ONEPAT(DIC,IEN,FIRST) ; Prompts the user for a Patient  ; Input: DIC - Variable/Array of settings needed for ^DIC call  ; FIRST - Set to 1 initially and then 0 for subsequent calls  ; Output: FIRST - Set to 0  ; IEN - IEN of the Patient  ; null of no selection was made  S DIC("A")=$S(FIRST:"Select Patient: ",1:"Select Another Patient: ")  D ^DIC  S FIRST=0,IEN=Y  Q  ; |
| SHOWFILT(FILTERS) ; Display  ; Displays the currently selected filter selections for the  ; Billing and NVC Precert Worklist display  ; Input: FILTERS() - Array of filter settings. See FILTERS for a detailed  ; explanation of the FILTERS array  ; Output: Current Filter settings are displayed  ;  N DIR,DIROUT,DIRUT,DTOUT,DUOUT,IEN,LEN,IBXX,IBY,IBZ  I IBWLTYP="B" D  . W !!!,"Type of Review: "  . S IBY=$P(FILTERS(0),U,1)  . W $S(IBY=1:"Facility Revenue",IBY=2:"RUR SC",IBY=3:"Billing",1:"")  . S IBZ=$P(FILTERS(0),U,4) W " ("\_$S(IBZ=1:"First Party Copay",1:"Third Party")\_")"  I IBWLTYP="P" D  . W !!!,"Pre-certification Department: "  . S IBY=$P(FILTERS(0),U,1)  . W $S(IBY=1:"Insurance Verification",IBY=2:"RUR",1:"")  ;  W !,"Show All Divisions or Selected Divisions: "  W $S($P(FILTERS(0),U,2)=0:"All",1:"Selected")  ;  ; Division list (if any)  I ($P(FILTERS(0),U,2)=1) D  . W !,"Divisions to Display: "  . S LEN=20,IEN=0  . F S IEN=$O(FILTERS(1,IEN)) Q:IEN="" D  . . S IBXX=$$GET1^DIQ(4,IEN\_",",.01)  . . S LEN=LEN+$L(IBXX)  . . I LEN+2<80 D Q  . . . W IBXX  . . . I $O(FILTERS(1,IEN))'="" D  . . . . S LEN=LEN+2  . . . . W ", "  . . S LEN=20  . . W !," ",IBXX  ; |
| W !,"All Patients or Selected Patients: "  W $S($P(FILTERS(0),U,3)=0:"All",1:"Selected")  ; Patient Inclusion list (if any)  I ($P(FILTERS(0),U,3)=1) D  . W !,"Patients to Display: "  . S LEN=20,IEN=0  . F S IEN=$O(FILTERS(2,IEN)) Q:IEN="" D  . . S IBXX=$$GET1^DIQ(2,IEN\_",",.01)  . . S LEN=LEN+$L(IBXX)  . . I LEN+2<80 D Q  . . . W IBXX  . . . I $O(FILTERS(2,IEN))'="" D  . . . . S LEN=LEN+2  . . . . W ", "  . . S LEN=20  . . W !," ",IBXX  ;  K DIR  D PAUSE^VALM1  Q  ;  EXPAND ; Expand Item (EE)  D FULL^VALM1  N I,J,DFN,IBFBA,IBXX,VALMY,ECNT,IBAUTH,IBNAME  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F S IBXX=$O(VALMY(IBXX)) Q:'IBXX D  . K ^TMP("IBFBWE",$J)  . S ECNT=$G(^TMP("IBFBWLX",$J,IBXX))  . S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(ECNT,U,4)  . S ^TMP("IBFBWE",$J)=DFN\_U\_IBNAME\_U\_IBAUTH\_U\_IBFBA  . D EN^VALM("IB BILLING WORKLIST EXPAND")  . Q  K ^TMP("IBFBWE",$J)  S VALMBCK="R"  Q  ; |
| LINKCT ; Claims Tracking (CT)  I IBWLTYP="P",IBGRP=1 D Q 🡸Not called in Billing Worklist  . W !," This action not available for IV queue."  . D PAUSE^VALM1  . K ^TMP($J,"IBCLMTRK")  . S VALMBCK="R"  D FULL^VALM1  K ^TMP($J,"IBCLMTRK")  N I,J,CTDT,CTIEN,CTLN1,CTUSR,DFN,ECNT,GOTPAT,IBFBA,IBAUTH,IBEND,IBNAME,IBST,IBXX,VALMY  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F S IBXX=$O(VALMY(IBXX)) Q:'IBXX D  . S ECNT=$G(^TMP("IBFBWLX",$J,IBXX))  . S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(ECNT,U,4)GOTPAT=1  . D GETDTS^IBFBUTIL(IBIEN)  . I IBEND="" S IBEND="3991231"  . S ^TMP($J,"IBCLMTRK")=DFN\_U\_IBST\_U\_IBEND\_U\_IBAUTH\_U\_IBFBA  . D EN^VALM("IBT CLAIMS TRACKING EDITOR") |
| I IBWLTYP="P",$D(D0) D 🡸Not called in Billing Worklist  . S CTIEN=D0  . I '$D(^IBT(356,CTIEN)) Q  . S CTLN1=^IBT(356,CTIEN,1)  . S CTDT=$P($P(CTLN1,U,1),".",1)  . S CTUSR=$P(CTLN1,U,2)  . I $G(DUZ)=CTUSR,DT=CTDT D  .. N Y,X  .. W !!  .. S DIR("A")="Link last Claims Tracking entry to current auth for "\_IBNAME\_"? "  .. S DIR("?")="Please answer Yes or No."  .. S DIR("B")="YES",DIR(0)="YA^^"  .. D ^DIR K DIR  .. I Y(0)'="YES" Q  .. S CTTMP=^TMP($J,"IBCLMTRK")  .. S DFN=$P(CTTMP,U,1)  .. S IBAUTH=$P(CTTMP,U,4)  .. S IBXX=""  .. S IBXX=$O(^IBFB(360,"D",DFN,IBAUTH,IBXX))  .. S $P(^IBFB(360,IBXX,1),U,1)=CTIEN  K ^TMP($J,"IBCLMTRK")  S VALMBCK="R"  Q  ; |
| ACTIONS ; Worklist Action (WA)  D FULL^VALM1  N I,J,DFN,IBFBA,IBXX,VALMY,ECNT,IBAUTH,IBNAME  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F S IBXX=$O(VALMY(IBXX)) Q:'IBXX D  . K ^TMP("IBFBWA",$J)  . S ECNT=$G(^TMP("IBFBWLX",$J,IBXX))  . S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(ECNT,U,4)  . S ^TMP("IBFBWA",$J)=DFN\_U\_IBNAME\_U\_IBAUTH\_U\_IBFBA  . I IBWLTYP="B" D 🡸Billing Worklist code  .. D EN^VALM("IB BILLING WORKLIST ACTIONS")  . I IBWLTYP="P" D  .. I IBGRP=1 D EN^VALM("IB NVC PRECERT WORKLIST IV")  .. I IBGRP=2 D EN^VALM("IB NVC PRECERT WORKLIST RUR")  K ^TMP("IBFBWA",$J)  K ^TMP("VALMAR",$J)  I IBWLTYP="B" D BLDWL^IBFBWL1  I IBWLTYP="P" D BLDWL^IBFBWL5  S VALMBCK="R"  Q  ; |
| HISTORY ; Worklist History (HI)  D FULL^VALM1  N I,J,DFN,ECNT,IBA,IBAUTH,IBB,IBHDT,IBHLG,IBHUSR,IBNAME,IBNAME,IBY,IBX,IBXX,VALMY  D EN^VALM2($G(XQORNOD(0)))  I $D(VALMY) S IBXX=0 F S IBXX=$O(VALMY(IBXX)) Q:'IBXX D  . K ^TMP("IBFBWH",$J)  . S ECNT=$G(^TMP("IBFBWLX",$J,IBXX))  . S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(ECNT,U,4)  . I IBFBA’=”” S IBY=IBFBA  . I IBFBA=”” D  .. S IBX="" F S IBX=$O(^IBFB(360,"C",DFN,IBX)) Q:IBX="" D  ... I $P(^IBFB(360,IBX,0),U,3)=IBAUTH S IBY=IBX  . S IBA=0 F S IBA=$O(^IBFB(360,IBY,4,IBA)) Q:IBA="" D  .. S IBHDT=$$FDATE^VALM1($P(^IBFB(360,IBY,4,IBA,0),U,1))  .. S IBHLG=$P(^IBFB(360,IBY,4,IBA,0),U,2)  .. S IBHUSR=$P(^IBFB(360,IBY,4,IBA,0),U,3)  .. S ^TMP("IBFBWH",$J,IBA)=IBHDT\_U\_IBHLG\_U\_IBHUSR  . D EN^VALM("IB BILLING WORKLIST HISTORY")  . Q  K ^TMP("IBFBWH",$J)  S VALMBCK="R"  Q  ; |
| REFRESH ; Special Main Screen List Refresh 🡸Special refresh code was needed to redisplay list  K ^TMP("IBFBWL",$J)  I IBWLTYP="B" D  . D GETAUT^IBFBWL1(IBGRP)  . D BLDWL^IBFBWL1  I IBWLTYP="P" D  . D GETAUT^IBFBWL5(IBGRP)  . D BLDWL^IBFBWL5  S VALMBCK="R"  Q  ;  KILLGLB ; Kill Worklist Globals  K ^TMP("IBFBWL",$J)  K ^TMP("IBFBWLX",$J)  K ^TMP("IBFBWA",$J)  K ^TMP("IBFBWE",$J)  K ^TMP("IBFBWH",$J)  K ^TMP("VALMAR",$J)  K ^TMP("XQORS",$J)  D CLEAR^VALM1  Q  ;  CHKFILT ; Check Filters  I $P($G(FILTERS(0)),U,2)=0 S IBDIVS="All"  I $P($G(FILTERS(0)),U,2)'=0 D  . S IBIEN=0,IBFST=1  . F S IBIEN=$O(FILTERS(1,IBIEN)) Q:IBIEN="" D  .. S IBXX=$E($$GET1^DIQ(4,IBIEN\_",",.01),1,15)  .. I IBFST S IBFST=0,IBDIVS=IBXX Q  .. S IBDIVS=IBDIVS\_","\_IBXX  S IBY=$P(FILTERS(0),U,1)  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ;  EXIT ; -- exit code  D KILLGLB  D CLEAN^VALM10  D ^%ZISC  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWL1 – Loop and create main Precert Worklist page  (Note: ^IBFBWL5 is NOT used for the Billing Worklist. It is used to create the main IB NVC Precert Worklist page, only, NSR #20081010) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.2, FRFEE-2.3, FRFEE-2.4,FRFEE-2.3-03, FRFEE-2.3-07 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^IBFBWL |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | FEE BASIS PATIENT AUTHORIZATION SUBFILE (#161.01)  IB-FB INTERFACE TRACKING FILE (#360)  INSURANCE COMPANY FILE (#36)  FEE BASIS CONTRACT FILE (#161.43) | | | | |
| **Related Protocols** | List  IB BILLING WORKLIST  Menu Protocol  IB BILLING WORKLIST MENU  Action Protocols  IB BILLING WORKLIST REFRESH - REFRESH^IBFBWL  IB BILLING WORKLIST EXPAND - EXPAND^IBFBWL  IB BILLING WORKLIST CLAIMS TRACKING – LINKCT^IBFBWL  IB BILLING WORKLIST ACTIONS – ACTIONS^IBFBWL  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes – in process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWL1 ;ALB/PAW-IB Billing Worklist Main ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;;  ;  GETAUT(IBGRP) ; Obtain new invoices, based upon review group  ;required input IBGRP = 1 (FR) or 2 (SC) or 3 (BI)  ;output ^TMP("IBFBWL",$J), containing auths for group queue  N IBA  S IBA=""  I IBGRP=1 D LOOP1 🡸Facility Revenue Queue  I IBGRP=2 D LOOP2 🡸RUR-SC Queue  I IBGRP=3 D LOOP3 🡸Billing Queue  Q  ;  LOOP1 ; Loop to create Fee Revenue Worklist  F S IBA=$O(^IBFB(360,"FR","FR",IBA)) Q:IBA="" D  . D BLDTMP  Q  ;  LOOP2 ; Loop to create RUR SC/SA Worklist  F S IBA=$O(^IBFB(360,"SC","SC",IBA)) Q:IBA="" D  . D BLDTMP  Q  ;  LOOP3 ; Loop to create Billing Worklist  F S IBA=$O(^IBFB(360,"BI","BI",IBA)) Q:IBA="" D  . D BLDTMP  Q  ; |
| BLDTMP ; Build ^TMP("IBFBWL",$J)  N DFN,IBAUTH,IBFPTP,IBDIV,IBDOB,IBIEN,IBNAME,IBSSN,IBFP,IBST,IBEND  I '$D(^IBFB(360,IBA)) Q  S DFN=$$GET1^DIQ(360,IBA\_",",.02,"I")  S IBAUTH=$$GET1^DIQ(360,IBA\_",",.03)  S IBIEN=IBAUTH\_","\_DFN\_","  S IBDIV=$$GET1^DIQ(161.01,IBIEN,101,"I")  I IBDIV'="",$D(FILTERS(1)) I '$D(FILTERS(1,IBDIV)) Q ; If filtering by select divisions  I IBDIV="",$D(FILTERS(1)) Q ; Filtering by division, but no division on auth  I $D(FILTERS(2)) I '$D(FILTERS(2,DFN)) Q ; If filtering by select patients  S IBFPTP=$$GET1^DIQ(360,IBA\_",",3.02)  I $P(FILTERS(0),U,4)=1,IBFPTP'=1 Q ; If filtering by first party  I $P(FILTERS(0),U,4)=3,IBFPTP=1 Q ; If filtering by first party  D DEMOS  Q  ; |
| DEMOS ; Demographics  N IBFP,IBINV,IBSSN,IBST,IBSTK,IBSTL,IBSSX,IBSSLE,IBSSLS,VA,VADM,VAERR  D DEM^VADPT  I VAERR K VADM  S IBNAME=$G(VADM(1)) S:IBNAME="" IBNAME=" "  S IBDOB=$P($G(VADM(3)),U,1)  S IBSSX=$P($G(VADM(2)),U,1),IBSSLE=$L(IBSSX),IBSSLS=6 I $E(IBSSX,IBSSLE)="P" S IBSSLS=5  S IBSSN=$E(IBNAME,1)\_$E(IBSSX,IBSSLS,IBSSLE)  S IBFP=$$GET1^DIQ(161.01,IBIEN\_",",.03) ; NVC  I IBFP="" S IBFP="UNK"  S IBST=""  D GETST^IBFBUTIL(IBA)  I IBST="" S IBST="UNK"  ; Sort by DOS (primary), Type (secondary)  S ^TMP("IBFBWL",$J,IBST,IBFP,IBNAME,DFN,IBAUTH,IBA)=IBNAME\_U\_IBDOB\_U\_IBSSN\_  U\_IBFP\_U\_IBST\_U\_IBINV  Q  ; |
| BLDWL ; Build Work List Screen  ; build display lines  K ^TMP("IBFBWLX",$J)  N DFN,IBAUTH,IBFP,IBNAME,IBST,IBXX,FIRST,LINE,VCNT  S (VALMCNT,FIRST,VCNT,CNT)=0  S IBST=""  F S IBST=$O(^TMP("IBFBWL",$J,IBST)) Q:IBST="" D  . S IBFP=""  . F S IBFP=$O(^TMP("IBFBWL",$J,IBST,IBFP)) Q:IBFP="" D  .. S IBNAME=""  .. F S IBNAME=$O(^TMP("IBFBWL",$J,IBST,IBFP,IBNAME)) Q:IBNAME="" D  ... S FIRST=1  ... S DFN=""  ... F S DFN=$O(^TMP("IBFBWL",$J,IBST,IBFP,IBNAME,DFN)) Q:DFN="" D  .... S IBAUTH=""  .... F S IBAUTH=$O(^TMP("IBFBWL",$J,IBST,IBFP,IBNAME,DFN,IBAUTH)) Q:IBAUTH="" D  ….. S IBA=””  ….. F S IBG=$O(^TMP(“IBFBWL”,$J,IBST,IBFP,IBNAME,DFN,IBAUTH,IBA)) Q:IBA=”” D  ...... S VCNT=VCNT+1  ...... S LINE=$$SETL("",VCNT,"",1,4) ;line#  ...... S IBXX=^TMP("IBFBWL",$J,IBST,IBFP,IBNAME,DFN,IBAUTH,IBA)  …... S IBNAME=$P(IBXX,U)  …... S LINE=$$SETL(LINE,IBNAME,"",5,23)  ...... S LINE=$$SETL(LINE,$$FDATE^VALM1($P(IBXX,U,2)),"",28,8)  …... S LINE=$$SETL(LINE,$P(IBXX,U,3),"",37,5)  ...... S LINE=$$SETL(LINE,$P(IBXX,U,4),"",43,10)  …... I $P(IBXX,U,5)'="UNK" S LINE=$$SETL(LINE,$$FDATE^VALM1($P(IBXX,U,5)),"",54,8)  ...... E S LINE=$$SETL(LINE,"","",54,8)  …... S LINE=$$SETL(LINE,$P(IBXX,U,6),"",63,16)  ...... S VALMCNT=VALMCNT+1  …... D SET^VALM10(VALMCNT,LINE,VCNT)  ...... S ^TMP("IBFBWLX",$J,VCNT)=DFN\_U\_IBNAME\_U\_IBAUTH\_U\_IBA  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 |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWL2 – Expand Worklist Item  (Note: The Billing Worklist shares this routine with the Precert Worklist. Many fields display on the Billing Worklist only.) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.3-01 through FRFEE-2.3-08, FBFEE-2.4,FRFEE-2.8 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called from IB BILLING WORKLIST Action “EE” – EXPAND^IBFBWL | GETAUTH^IBFBUTIL  GETPAY^IBFBUTIL  CL^IBACV (special auth elig)  CVEDT^IBACV (combat vet elig) ALL^IBCNS1 DAT1^IBOUTL |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | PATIENT FILE (#2)  IB-FB INTERFACE TRACKING FILE (#360)  FEE BASIS VENDOR FILE (#161.2)  NEW PERSON FILE (#200)  DISABILITY CONDITION FILE (#31)  INSURANCE COMPANY FILE (#36)  FEE BASIS PATIENT FILE AUTHORIZATION SUBFILE (#161.01) | | | | |
| **Related Protocols** | Lists  IB BILLING WORKLIST MENU  IB BILLING WORKLIST EXPAND - EXPAND^IBFBWL  Corresponding Menu (Protocol)  IB BILLING WORKLIST EXPAND MENU PROTOCOL  Action Protocols  IB BILLING WORKLIST EXPAND REFRESH – REFRESH^IBFBWL2 | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes – in process | | | | |
| **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 |
| **New Routine** |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWL2 ;ALB/PAW-IB BILLING Worklist Expand Item ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;;  EN ; -- main entry point for BILLING WORKLIST EXPAND  D EN^VALM("IB PRECERT WORKLIST EXPAND")  Q  ;  HDR ; -- header code  S VALM("TITLE")=" Expanded NVC"  Q  ;  INIT ; -- init variables and list array  ; input - ^TMP("IBFBWE",$J)=DFN^IBNAME^IBAUTH  ; output - Expanded worklist screen  I '$D(^TMP("IBFBWE",$J)) Q  N CNT,DFN,ECNT,IBFBA,IBNAME,IBAUTH,IBST,LINE,VAEL  S ECNT=$G(^TMP("IBFBWE",$J))  S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(EC)  D BLD  D GETSC  D GETSTA  D GETINS  D BLDEXP  S VALMBCK="R"  Q  ; |
| BLD ; build data to display  N IBARRAY,IB7078,IBCP,IBIEN,IBDX1,IBDX2,IBDX3,IBEND,IBENDX,IBFILL,IBFPI  N IBPUR,IBREF,IBREFA,IBREM,IBREM1,IBREMARK,IBRET,IBSS,IBSTX,IBTAX,IBVNE  N IBSSLS,IBVTAX,IBX,IBXX,VA  S CNT=0  D ELIG^VADPT  S IBSSX=$$GET1^DIQ(2,DFN\_",",.09,"I"),IBSSLE=$L(IBSSX),IBSSLS=6 I $E(I5  S IBSS=$E(IBNAME,1)\_$E(IBSSX,IBSSLS,IBSSLE)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="PATIENT "  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="SSN X  S IBIEN=IBAUTH\_","\_DFN\_","  D GETAUTH^IBFBUTIL(IBIEN,"IBARRAY") ; Call API for Authorization Sub-s  I IBWLTYP="P" D  . S IBST=$G(IBARRAY(161.01,IBIEN,.01,"I"))  . S IBEND=$G(IBARRAY(161.01,IBIEN,.02,"I"))  . S IBSTX=$G(IBARRAY(161.01,IBIEN,.01,"E"))  . S IBENDX=$G(IBARRAY(161.01,IBIEN,.02,"E"))  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="AUTH DOS RANGE X  I IBWLTYP="B" D  . S (IBST,IBINV)=""  . I IBFBA'="" S IBXX=IBFBA  . I IBFBA="" D  .. S IBX="" F S IBX=$O(^IBFB(360,"C",DFN,IBX)) Q:IBX="" D  ... I $$GET1^DIQ(360,IBX\_",",.03)=IBAUTH S IBXX=IBX  . D GETST^IBFBUTIL(IBXX)  . D GETPAY^IBFBUTIL(IBXX)  . I IBST'="" S IBSTX=$$FDATE^VALM1(IBST)  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="INVOICE NVC )  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="INVOICE# V  . I IBFPNO=2!(IBFPNO=3) S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH)  . I IBFPNO=6!(IBFPNO=7) S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH)  . ; I IBFPNO=3 S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="VE)  . I $G(IBINV)'="" D  .. S IBPAYX=""  .. F S IBPAYX=$O(IBRET(162.03,IBPAYX)) Q:IBPAYX="" D  ... S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" SERVICE CPT)  ... I $D(IBRET(162.03,IBPAYX,2,"I")) S IBINVPD=$FN(IBRET(162.03,IBPAYX)  ... S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" AMOUNT PAID)  ... S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" RENDERING P)  ... S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" RENDERING N)  ... S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" RENDERING T) |
| S IBFP=$G(IBARRAY(161.01,IBIEN,.03,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="AUTHORIZATION NVCP  S IBPUR=$G(IBARRAY(161.01,IBIEN,.07,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="PURPOSE OF VISIT R  S IBVND=$G(IBARRAY(161.01,IBIEN,.04,"E"))  S IBVNDA=$G(IBARRAY(161.01,IBIEN,.04,"I"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="NON-VA LOCATION D  S IBVNPI=""  I IBVNDA'="" S IBVNPI=$$GET1^DIQ(161.2,IBVNDA\_",",41.01)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" NPI# I  S IBVTAX=""  I IBVNDA'="" S IBVTAX=$$GET1^DIQ(161.2,IBVNDA\_",",42)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" TAXONOMY X  S IB7078=$G(IBARRAY(161.01,IBIEN,.055,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="ASSOC 7078/583 8  S IBREF=$G(IBARRAY(161.01,IBIEN,104,"E"))  S IBREFA=$G(IBARRAY(161.01,IBIEN,104,"I"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="ORDERING PHYSICIAF  S IBNPI=""  I IBREFA'="" S IBNPI=$$GET1^DIQ(200,IBREFA\_",",41.99)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" NPI# I  ; Placeholder if VA starts to use Taxonomy for VA physicians  S IBTAX=""  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" TAXONOMY X  S IBMT=$P($G(VAEL(9)),U,2)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="MEANS TEST STATUST  D DISP^IBARXEU(DFN,IBST,1,"")  S IBCP=$P($G(X),U,2)  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="MED COPAY EXEMP SP  S IBDX1=$G(IBARRAY(161.01,IBIEN,.08,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="DIAGNOSIS 1 1  S IBDX2=$G(IBARRAY(161.01,IBIEN,.085,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="DIAGNOSIS 2 2  S IBDX3=$G(IBARRAY(161.01,IBIEN,.086,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="DIAGNOSIS 3 3  S IBICDDX=$G(IBARRAY(161.01,IBIEN,.087,"E"))  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="ICD DIAGNOSIS X  I '$D(IBARRAY(161.01,IBIEN,.021)) S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,"  S IBREM="",IBREM1=1  F S IBREM=$O(IBARRAY(161.01,IBIEN,.021,IBREM)) Q:IBREM=""!(IBREM="E")D  . S IBREMARK=IBARRAY(161.01,IBIEN,.021,IBREM)  . I IBREM1 S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="AUTH R0  . E S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" K  Q  ; |
| BLDEXP ; Build expand screen  D FULL^VALM1  N IBXX  S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3)  S VALMCNT=0  S IBXX=""  F S IBXX=$O(^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,IBXX)) Q:+IBXX=0 D  . S LINE=^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,IBXX)  . S VALMCNT=VALMCNT+1  . D SET^VALM10(VALMCNT,LINE,"")  S VALMCNT=VALMCNT+1  Q  ;  GETSC ; Get SC and STA  N IBD,IBI,IBX,IBY,IBSC,IBSC1,FIRST  S FIRST=1  I VAEL(3)=0 S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="SERVIQ  S IBSC="SERVICE CONNECTED : "\_$P(VAEL(3),U,2)\_"%"  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=IBSC  I '$O(^DPT(DFN,.372,0)) S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTHQ  S IBI=0 F S IBI=$O(^DPT(DFN,.372,IBI)) Q:'IBI D  . S IBX=$G(^DPT(DFN,.372,IBI,0)),IBY=$G(^DIC(31,+IBX,0))  . S IBD=$S($P(IBY,U,4)="":$P(IBY,U,1),1:$P(IBY,U,4))\_" ("\_$P(IBX,U,2)\_"  . I FIRST S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="RATED DQ  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" D  Q  ;  GETSTA ; Get Special Authority Eligibility  N IBY,FIRST,FOUND,IBADT,IBARR  S IBADT=IBST  S FOUND=0  S FIRST=1  D CL^IBACV(DFN,IBADT,"",.IBARR)  I $D(IBARR(3)) S IBY="SC TREATMENT" D GETSTA1  I $D(IBARR(7)),+$$CVEDT^IBACV(DFN,IBDAT) S IBY="COMBAT VETERAN" D GETS1  I $D(IBARR(1)) S IBY="AGENT ORANGE" D GETSTA1  I $D(IBARR(2)) S IBY="IONIZING RADIATION" D GETSTA1  I $D(IBARR(4)) S IBY="SOUTHWEST ASIA" D GETSTA1  I $D(IBARR(8)) S IBY="PROJECT 112/SHAD" D GETSTA1  I $D(IBARR(5)) S IBY="MILITARY SEXUAL TRAUMA" D GETSTA1  I $D(IBARR(6)) S IBY="HEAD/NECK CANCER" D GETSTA1  I 'FOUND S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="STA  Q  ;  GETSTA1 ; Set in ^TMP("IBFBWE",$J)  I FIRST S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="SPEC AUTH  S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" Y  Q  ; |
| GETINS ; Get insurance information  N EXDTCK,IBEFF,IBEXP,IBGRP,IBINSCO,IBINS0,IBINS1,IBINS7,IBSUB,IBVERBY,S  D ALL^IBCNS1(DFN,"IBINS")  S (IBX,IBINSYES)=0  I '$D(IBINS) S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="INSUQ  F S IBX=$O(IBINS(IBX)) Q:'IBX D  . S IBINS0=IBINS(IBX,0)  . S IBINS1=IBINS(IBX,1)  . S IBINS7=$G(IBINS(IBX,7))  . S IBINSCO=$P(^DIC(36,+IBINS0,0),U,1)  . S IBSUB=$P(IBINS7,U,1)  . S IBGRP=$P(IBINS0,U,3)  . S EXDTCK=+$P(IBINS0,U,4)  . I EXDTCK,EXDTCK<IBST Q ; if insurance expired before the from date t  . S IBINSYES=1  . S IBEFF=$$DAT1^IBOUTL($P(IBINS0,U,8))  . S IBEXP=$$DAT1^IBOUTL($P(IBINS0,U,4))  . S IBVERDT=$P($G(IBINS1),U,3)  . S IBVERBY=$P($G(IBINS1),U,4)  . I IBVERDT'="" S IBVERDT=$$FDATE^VALM1(IBVERDT)  . I IBVERBY'="" S IBVERBY="BY "\_$$GET1^DIQ(200,IBVERBY\_", ",.01)  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="INSURANCE O  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" SUBSCRIBER B  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" GROUP P  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" EFF DATE F  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" EXP DATE P  . S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)=" DT LAST VERIY  I IBINSYES=0 S CNT=CNT+1,^TMP("IBFBWE",$J,IBNAME,DFN,IBAUTH,CNT)="INSU"  Q  ; |
| REFRESH ; Special Expand Screen Refresh  K ^TMP("IBFBWE",$J)  D BLD  D GETSC  D GETSTA  D GETINS  D BLDEXP  S VALMBCK="R"  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ;  EXIT ; -- exit code  K ^TMP("IBFBWE",$J)  D ^%ZISC  S VALMBCK="R" Q  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWL3 - Billing Worklist Actions (Note: ^IBFBWL6 is not used for the Billing Worklist. It is for IB NVC Precert Worklist Actions, only – NSR #20081010. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.6 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by IB BILLING WORKIST Action “WA” – ACTIONS^IBFBWL | Standard ListMan routines  ^DIQ ^DIE  %DTC DIM^IBYMTC |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | PATIENT FILE (#2) | | | | |
| **Related Protocols** | Lists  IB BILLING WORKLIST MENU  IB BILLING WORKLIST ACTIONS – ACTIONS^IBFBWL  Corresponding Menu (Protocol)  IB BILLING WORKLIST ACTION MENU  Action Protocols  IB BILLING WORKLIST COMPLETE  IB BILLING WORKLIST REMOVE | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes – in process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWL3 ;ALB/PAW-IB BILLING Worklist Actions ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;;  EN ; -- main entry point for IB BILLING WORKLIST ACTIONS  ; add code to do filters here  ;  D EN^VALM("IB BILLING WORKLIST ACTIONS")  Q  ;  HDR ; -- header code  ;  N IBSS,IBSSX,IBSSLE,IBSSLS  S VALM("TITLE")=" Worklist Actions"  S IBSSX=$$GET1^DIQ(2,DFN,.09,"I"),IBSSLE=$L(IBSSX),IBSSLS=6 I $E(IBSSX5  S IBSS=$E(IBNAME,1)\_$E(IBSSX,IBSSLS,IBSSLE)  S VALMHDR(2)=" PATIENT: "\_IBNAME\_" (ID: "\_IBSS\_")"  Q  ; |
| INIT ; -- init variables and list array  ; input - ^TMP("IBFBWA",$J)=DFN^IBNAME^IBAUTH  ; output - none  N DFN,ECNT,IBAUTH,IBFBA,IBNAME  I '$D(^TMP("IBFBWA",$J)) Q  S ECNT=$G(^TMP("IBFBWA",$J))  S DFN=$P(ECNT,U,1),IBNAME=$P(ECNT,U,2),IBAUTH=$P(ECNT,U,3),IBFBA=$P(EC)  D BLD  Q  ;  BLD ; Build data to display  N IBGRPX,VALMY  D FULL^VALM1  S IBGRPX=$S(IBGRP=1:"Facility Revenue Review",IBGRP=2:"RUR SC/SA Revie)  D SET^VALM10(1,"","")  D SET^VALM10(2," Available Actions:")  D SET^VALM10(3,"","")  D SET^VALM10(4," Enter 1 to COMPLETE the "\_IBGRPX\_" process.")  D SET^VALM10(5," Enter 2 to REMOVE an item from the worklist.")  I IBGRP=2 D RURRC^IBFBWL6  Q  ; |
| DONE ; Review is complete (for IBGRP)  N IBEVENT,IBIEN,IBSCSA,IENROOT,FDA  S IENROOT=""  D FIND  I IBGRP=2 D RURRC^IBFBWL6 ; Additional prompt for RUR reason codes  I IBGRP=1 D  . D SCSA ; Determine if Service Connected or Special Treatment Authors  . S FDA(360,IBIEN\_",",2.03)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  . I IBSCSA D ; If SC/STA move to RUR-SC queue  .. S FDA(360,IBIEN\_",",2.04)="SC"  . I 'IBSCSA D ; If no SC/STA move to billing queue  .. S FDA(360,IBIEN\_",",2.05)="BI"  . D UPDATE^DIE("","FDA","IENROOT")  I IBGRP=2 D  . S FDA(360,IBIEN\_",",2.04)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  . S FDA(360,IBIEN\_",",2.05)="BI"  . D UPDATE^DIE("","FDA","IENROOT")  I IBGRP=3 D  . S FDA(360,IBIEN\_",",2.05)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  . D BILLING  D RESET  I IBGRP=2 D RURRCPR^IBFBWL6  S IBEVENT=$S(IBGRP=1:"Fac Rev",IBGRP=2:"RUR-SC/SA",1:"Billing")\_"-Comp)  D LOGUPD  S IBGRPX=$S(IBGRP=1:"Facility Revenue Review",IBGRP=2:"RUR SC/SA Revie)  W !," Item for "\_IBNAME\_" has completed "\_IBGRPX\_"."  D PAUSE^VALM1  S VALMBCK="R"  Q  ;  REM ; Remove Item from Worklist (log IBGRP)  N IBEVENT,IBIEN,IENROOT  S IENROOT=""  D FIND  I IBGRP=2 D RURRC^IBFBWL6 ; Additional prompt for RUR reason codes  I IBGRP=1 D  . S FDA(360,IBIEN\_",",2.03)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  I IBGRP=2 D  . S FDA(360,IBIEN\_",",2.04)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  I IBGRP=3 D  . S FDA(360,IBIEN\_",",2.05)="XX"  . D UPDATE^DIE("","FDA","IENROOT")  D RESET  S IBEVENT=$S(IBGRP=1:"Fac Rev",IBGRP=2:"RUR-SC/SA",1:"Billing")\_"-Item"  D LOGUPD  W !," Item for "\_IBNAME\_" has been removed from the worklist."  W !," Please update Claims Tracking with Non-billable Reason, if neede"  D PAUSE^VALM1  S VALMBCK="R"  Q  ; |
| FIND ; Find Auth Match  I IBFBA'="" S IBIEN=IBFBA Q  N IBX  S IBX="" F S IBX=$O(^IBFB(360,"C",DFN,IBX)) Q:IBX="" D  . I $P(^IBFB(360,IBX,0),U,3)=IBAUTH S IBIEN=IBX  Q  ;  LOGUPD ; Update Log  N FDA,IBDT,IBLOG  S IBDT=$$NOW^XLFDT()  S FDA(360.04,"+1,"\_IBIEN\_",",.01)=IBDT,FDA(360.04,"+1,"\_IBIEN\_",",.03)Z  S IBLOG=$P($G(^IBFB(360,IBIEN,4,0)),U,3)  S IBLOG=IBLOG+1  S FDA(360.04,"+1,"\_IBIEN\_",",.02)=IBEVENT  D UPDATE^DIE("","FDA")  S ^IBFB(360,"DFN",DFN,DT,IBIEN,IBLOG)=""  S ^IBFB(360,"DT",DT,DFN,IBIEN,IBLOG)=""  Q  ;  SCSA ; Determine Service Connected or Special Authority Eligibility Status  N IBARR,IBSC,IBSTA,VAEL  S (IBSC,IBSCSA,IBSTA)=1  D ELIG^VADPT  I VAEL(3)=0 S IBSC=0  D GETST^IBFBUTIL(IBIEN)  D CL^IBACV(DFN,IBST,"",.IBARR)  I '$D(IBARR) S IBSTA=0  I 'IBSC,'IBSTA S IBSCSA=0  Q  ; |
| RESET ; Reset ^TMP global  N IBDOS,IBTYP  S IBDOS=""  F S IBDOS=$O(^TMP("IBFBWL",$J,IBDOS)) Q:IBDOS="" D  . S IBTYP=""  . F S IBTYP=$O(^TMP("IBFBWL",$J,IBDOS,IBTYP)) Q:IBTYP="" D  .. I $D(^TMP("IBFBWL",$J,IBDOS,IBTYP,IBNAME,DFN,IBAUTH,IBFBA)) D  ... K ^TMP("IBFBWL",$J,IBDOS,IBTYP,IBNAME,DFN,IBAUTH,IBFBA)  Q  ;  BILLING ; After final review by billing department, prepare to bill  N IBBC,IBDD,IBFPNUM,IBIFN,IBIDS,IBLOC,IBNPI,IBPAID,IBPAYX,IBREND,IBRET, IBRT,IBSER,IBSVC,IBSITE,IBST,IBTAX,PRCASV  N IBFBVND,IBA,IBHIT,IBIBANPI  D DEM^VADPT  D GETST^IBFBUTIL(IBIEN) ; Get Invoice, Start Date, Fee Program  I '$D(IBFPNUM) Q  S IBIDS(".03")=$G(IBST) ; Start Date of Care  S IBLOC=$S(IBFPNUM=7:2,1:1)  S IBIDS(".04")=IBLOC ; Location of Care 1 Hospital 2 Skilled Nursing  S IBBC=$S(IBFPNUM=2:3,IBFPNUM=3:3,1:1)  S IBIDS(".05")=IBBC ; Bill Classification 1 Inpatient 3 Outpatient  S IBIDS(".06")=1 ; Timeframe of Bill Set to 1 Admit through Discharge  S IBRT=""  S IBRT=$O(^DGCR(399.3,"B","FEE REIMB INS",IBRT))  S IBIDS(".07")=IBRT ; Rate Type Must be Fee Reimbursable Insurance  S IBIDS(".11")="i" ; Whos Responsible This is always set to "i" initially  S IBDD=$P($G(^IBE(350.9,1,1)),"^",25)  S IBIDS(".22")=IBDD ; Default Division - From IB Site Parameter File  S IBIDS(".27")="" ; Bill Charge Type - This is always set to null initially  S IBIDS("151")=$G(IBST) ; Statement Covers From Date  S IBIDS("152")=$G(IBST) ; Statement Covers To Date  S IBIDS("155")=0 ; Sensitive Record - 0 is No  S IBSER=$P(^IBE(350.9,1,1),U,14)  S PRCASV("SER")=IBSER ; MAS Service Pointer - From IB Site Parameter File  D GETPAY^IBFBUTIL(IBIEN)  S IBPAYX=""  S IBPAYX=$O(IBRET(162.03,IBPAYX))  S IBSITE=IBRET(162.03,IBPAYX,26,"I")  S PRCASV("SITE")=IBSITE ; Site  D ^IBCA2 ; This call completes initial bill and AR set up  S IBBILL=$P($G(IBDR("0")),U,1)  S FDA(360,IBIEN\_",",1.02)=IBBILL  D UPDATE^DIE("","FDA","IENROOT")  ;  S IBIFN=""  S IBIFN=$O(^DGCR(399,"B",IBBILL,IBIFN))  S IBNPI=IBRET(162.03,IBPAYX,64,"I") ; Non-VA Care Facility NPI  I IBNPI="" D ; See if NPI can be found via Auth and FB side  . S IBFBVND=$$GET1^DIQ(161.01,IBAUTH\_","\_DFN\_",",.04,"I")  . I IBFBVND'="" S IBNPI=$$GET1^DIQ(161.2,IBFBVND\_",",41.01,"I")  I IBNPI'="" D  . S (IBIBA,IBHIT)=""  . F S IBIBA=$O(^IBA(355.93,IBIBA)) Q:IBIBA="" D  .. S IBIBANPI=""  .. F S IBIBANPI=$O(^IBA(355.93,IBIBA,"NPISTATUS","C",IBIBANPI)) Q:IBIB ANPI=""!(IBHIT) D  ... I IBIBANPI=IBNPI S IBHIT=1 D  .... S FDA(399,IBIFN\_",",232)=IBIBA  .... D UPDATE^DIE("","FDA")  S FDA(399,IBIFN\_",",161)=30  D UPDATE^DIE("","FDA")  ;  ; Need to add billing provider and primary dx, if possible  ; 252 BILLING PROVIDER TAXONOMY <-Pntr [\*P8932.1']  ;  W !,"Bill "\_IBBILL\_" created for "\_IBNAME\_"."  D PAUSE^VALM1  Q  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ;  EXIT ; -- exit code  D ^%ZISC  S VALMBCK="R"  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWL4 – Worklist History | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.5 | | | |
| **Related Options** | IB BILLING WORKLIST | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by IB BILLING WORKLIST Action “HI” – HISTORY^IBFBWL | Standard ListMan routines ^DIQ |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | PATIENT FILE (#2)  NEW PERSON FILE (#200) IB-FB INTERFACE TRACKING FILE (#360) | | | | |
| **Related Protocols** | Lists  IB BILLING WORKLIST  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL  Corresponding Menu (Protocol)  None | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes – in process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWL4 ;ALB/PAW-IB BILLING Worklist Worklist History ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;;  EN ; -- main entry point for IB BILLING WORKLIST HISTORY  D EN^VALM("IB BILLING WORKLIST HISTORY")  Q  ;  HDR ; -- header code  N IBSS,IBSSX,IBSSLE,IBSSLS  S VALM("TITLE")=" Worklist History"  S IBSSX=$$GET1^DIQ(2,DFN\_",",.09,"I"),IBSSLE=$L(IBSSX),IBSSLS=6 I $E(IB  SSX,IBSSLE)="P" S IBSSLS=5  S IBSS=$E(IBNAME,1)\_$E(IBSSX,IBSSLS,IBSSLE)  S VALMHDR(2)=" PATIENT: "\_IBNAME\_" (ID: "\_IBSS\_")"  Q  ;  INIT ; -- init variables and list array  ; input - ^TMP("IBFBWH",$J,IBA)=IBHDT^IBHLG^IBHUSR  ; output - Worklist History Screen for one Patient / Auth  N IBA,IBHDT,IBHLG,IBLN,IBRUR,IBRURT,IBUSR,LINE,VCNT  S (VCNT,VALMCNT)=0  S IBA=""  F S IBA=$O(^TMP("IBFBWH",$J,IBA)) Q:+IBA=0 D  . S IBRURT=""  . S IBLN=^TMP("IBFBWH",$J,IBA)  . S IBHDT=$P(IBLN,U,1)  . S IBHLG=$P($P(IBLN,U,2),"|")  . S IBRUR=$P($P(IBLN,U,2),"|",2)  . I IBRUR'="" D RUR  . S IBUSR=$P(IBLN,U,3)  . I IBUSR="" S IBUSR="SYSTEM"  . E S IBUSR=$$GET1^DIQ(200,IBUSR\_",",.01)  . S VCNT=VCNT+1  . S LINE=$$SETL("",VCNT,"",1,4) ;line#  . D BLD  Q  ; |
| RUR ; Determine RUR Reason Code  S IBRURT=$S(IBRUR=1:"Pend Payer Action",IBRUR=2:"Addl Info Req-FR",IBRUR=3:"Auth Not Req - SC/SA",IBRUR=4:"AuthNotReq-PayerCont",1:"")  Q:IBRURT'=""  S IBRURT=$S(IBRUR=5:"Auth Not Reqd",IBRUR=6:"Auth Obtained",IBRUR=7:"Cont Stay Rev",IBRUR=8:"Discharge Rev Req",1:"")  Q:IBRURT'=""  S IBRURT=$S(IBRUR=9:"Part SC-Auth Worked",IBRUR=10:"PartStay/VisitAppd",IBRUR=11:"Auth Denied",1:"")  Q:IBRURT'=""  S IBRURT=$S(IBRUR=12:"AuthNotObt/NoROI/FR",IBRUR=13:"Related to Legal",1:"")  Q  ;  BLD ; build data to display  S LINE=$$SETL(LINE,IBHDT,"",5,8)  S LINE=$$SETL(LINE,IBHLG,"",14,20)  S LINE=$$SETL(LINE,IBRURT,"",35,20)  S LINE=$$SETL(LINE,IBUSR,"",56,23)  S VALMCNT=VALMCNT+1  D SET^VALM10(VALMCNT,LINE,VCNT)  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  ;  HELP ; -- help code  S X="?" D DISP^XQORM1 W !!  Q  ;  EXIT ; -- exit code  D ^%ZISC  S VALMBCK="R" Q  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ^IBFBWLR– Prints Historical Worklist Action Data by Patient or Date Range | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | FRFEE-2.5, FRFEE-2.10 | | | |
| **Related Options** | IB NVC BILLING WORKLIST REPORT | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by VistA Option: IB NVC BILLING WORKLIST REPORT |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | PATIENT FILE (#2)  NEW PERSON FILE (#200) IB-FB INTERFACE TRACKING FILE (#360) | | | | |
| **Related Protocols** | None | | | | |
| **Related Integration Control Registrations (ICRs)** | Yes – in process | | | | |
| **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 |
| --- |
| New Routine |

| Modified Logic (Changes are in bold) |
| --- |
| IBFBWLR ;ALB/PAW-NVC and Billing Worklist Worklist History Report ; 30-SEP-2015  ;;2.0;INTEGRATED BILLING;\*\*554\*\*;21-MAR-94;Build 3  ;;Per VA Directive 6402, this routine should not be modified.  ;;  EN ; -- Main entry point for NVC and Billing Worklist History Report  N DFN,IBC,IBDA,IBDB,IBDC,IBDL,IBDT,IBDTR,IBDT1,IBDT2,IBDTTM  N IBDUZ,IBEVNT,IBHDT,IBI,IBPG,IBQUIT,IBRANGE  D PROMPT  D PRINT  D EXIT  Q  ;  PROMPT ; - Report prompts  ; Can be run by PATIENT or DATE RANGE  S DIR(0)="S^P:Patient;D:Date Range"  S DIR("A")="Report by Patient or Date Range"  S DIR("B")="Date Range"  S DIR("?",1)="Enter P to print the worklist history data for one patient."  S DIR("?",2)="Enter D to print all worklist history data for a date range."  S DIR("?")="Enter a code from the list."  D ^DIR K DIR G:$D(DIRUT) EXIT  S IBRANGE=$S(Y="D":1,1:0)  ; |
| I IBRANGE D G:$D(DIRUT) EXIT  . ; Ask dates  . S DIR(0)="D^::EX",DIR("A")="From Date"  . ; Default from date is first day of current month  . S DIR("B")=$$FMTE^XLFDT($E(DT,1,5)\_"01")  . D ^DIR K DIR Q:$D(DIRUT)  . S IBDT1=Y  . S DIR(0)="DA^"\_IBDT1\_"::EX",DIR("A")="To Date: "  . ; Default to date is last day of specified month  . S X=IBDT1 D DIM^IBYMTC  . S DIR("B")=$$FMTE^XLFDT($E(IBDT1,1,5)\_X)  . D ^DIR K DIR Q:$D(DIRUT)  . S IBDT2=Y  ;  ; If not date range then ask patient  I 'IBRANGE D G:$D(DIRUT) EXIT  . S DIC=2,DIC(0)="AE"  . S DIC("A")="Select Patient: "  . D ^DIC  . S DFN=$P(Y,U)  ;  ; Ask device  S %ZIS="QM" D ^%ZIS G:POP EXIT  I $D(IO("Q")) D G EXIT  . S ZTRTN="QEN^IBFBWLR",ZTDESC="NVC/Billing Worklist History"  . F IBX="IBAAIN","IBDT\*","IBRANGE" S ZTSAVE(IBX)=""  . D ^%ZTLOAD,HOME^%ZIS K ZTSK  QEN ; queued entry  U IO  Q  ; |
| PRINT ; Report data  S IBQUIT=0  S IBPG=0 D NOW^%DTC S Y=% D DD^%DT S IBDTR=Y  K IBDL S IBDL="",$P(IBDL,"-",IOM)=""  ;  ; Build page header text for selection criteria  S:IBRANGE IBHDT(1)=" For "\_$$FMTE^XLFDT(IBDT1)\_" through "\_$$FMTE^XLFDT(IBDT2)  ;  D HD  ;  ; Initialize Counter  S IBC=0  ;  ; If by date range  I IBRANGE D  . S IBDT=IBDT1-.0000001  . F S IBDT=$O(^IBFB(360,"DT",IBDT)) Q:'IBDT!(IBDT>(IBDT2\_".999999")) D Q:IBQUIT  .. S IBDA="" F S IBDA=$O(^IBFB(360,"DT",IBDT,IBDA)) Q:'IBDA D Q:IBQUIT  ... S IBDB="" F S IBDB=$O(^IBFB(360,"DT",IBDT,IBDA,IBDB)) Q:'IBDB D Q:IBQUIT  .... S IBDC="" F S IBDC=$O(^IBFB(360,"DT",IBDT,IBDA,IBDB,IBDC)) Q:'IBDC D Q:IBQUIT  ..... S DFN=IBDA  ..... D SETVARS  ..... D PRINT1  ;  ; If by patient  I 'IBRANGE D  . S IBDA="" F S IBDA=$O(^IBFB(360,"DFN",DFN,IBDA)) Q:'IBDA D Q:IBQUIT  .. S IBDB="" F S IBDB=$O(^IBFB(360,"DFN",DFN,IBDA,IBDB)) Q:'IBDB D Q:IBQUIT  ... S IBDC="" F S IBDC=$O(^IBFB(360,"DFN",DFN,IBDA,IBDB,IBDC)) Q:'IBDC D Q:IBQUIT  .... D SETVARS  .... D PRINT1  ;  I IBC=0 W !,"No worklist history entries found."  ;  I IBQUIT W !!,"REPORT STOPPED AT USER REQUEST"  ;  I 'IBQUIT,$E(IOST,1,2)="C-" S DIR(0)="E" D ^DIR K DIR  D ^%ZISC  Q  ; |
| SETVARS ; Set variables  S IBDTTM=$P($G(^IBFB(360,IBDB,4,IBDC,0)),U,1)  S IBEVNT=$P($P($G(^IBFB(360,IBDB,4,IBDC,0)),U,2),"|")  S IBDUZ=$P($G(^IBFB(360,IBDB,4,IBDC,0)),U,3)  Q  ;  HD ; Page header  I $D(ZTQUEUED),$$S^%ZTLOAD S ZTSTOP=1,IBQUIT=1 Q  I $E(IOST,1,2)="C-",IBPG S DIR(0)="E" D ^DIR K DIR I 'Y S IBQUIT=1 Q  I $E(IOST,1,2)="C-"!IBPG W @IOF  S IBPG=IBPG+1  W !,"NVC/Billing Worklist History "  I IBRANGE W "by Date Range"  E W "by Patient"  W ?49,IBDTR,?72,"page ",IBPG  S IBI=0 F S IBI=$O(IBHDT(IBI)) Q:'IBI W !,IBHDT(IBI)  W !!,"Date/Time",?19,"Patient",?40,"Event",?61,"User"  W !,IBDL  Q  ;  PRINT1 ; Print one history record  N IBCNT,IBRUR,IBRURT,IBRURTX  S IBC=IBC+1  I $Y+9>IOSL D HD Q:IBQUIT  S Y=IBDTTM D DD^%DT S IBDTTM=Y  W !,$P(IBDTTM,":",1,2),?19,$E($$GET1^DIQ(2,DFN\_",",.01),1,20),?40,$E(IB  EVNT,1,20),?61,$E($$GET1^DIQ(200,IBDUZ\_",",.01),1,18)  I $P($P($G(^IBFB(360,IBDB,4,IBDC,0)),U,2),"|",2)'="" D  . S IBRURT=""  . S IBRUR=$P($P(^IBFB(360,IBDB,4,IBDC,0),U,2),"|",2)  . S IBRURT=$S(IBRUR=1:"Pend Payer Action",IBRUR=2:"Addl Info Req-FR",IB  RUR=3:"Auth Not Req - SC/SA",IBRUR=4:"AuthNotReq-PayerCont",1:"")  . Q:IBRURT'=""  . S IBRURT=$S(IBRUR=5:"Auth Not Reqd",IBRUR=6:"Auth Obtained",IBRUR=7:"  Cont Stay Rev",IBRUR=8:"Discharge Rev Req",1:"")  . Q:IBRURT'=""  . S IBRURT=$S(IBRUR=9:"Part SC-Auth Worked",IBRUR=10:"PartStay/VisitApp  d",IBRUR=11:"Auth Denied",1:"")  . Q:IBRURT'=""  . S IBRURT=$S(IBRUR=12:"AuthNotObt/NoROI/FR",IBRUR=13:"Related to Legal",1:"")  I $G(IBRURT)'="" W !?4,"RUR Reason: ",IBRURT  Q  ;  EXIT ;  I $D(ZTQUEUED) S ZTREQ="@"  K %,DIC,DIR,DIROUT,DIRUT,DTOUT,DUOUT,I,J,POP,X,Y  Q |

##### Templates

Table 5: Templates

| Templates | Description | | | |
| --- | --- | --- | --- | --- |
| Template Name | The Billing Worklist utilizes VistA List Manager “Lists” that are supported by “Protocols”. Some of the Worklists / Protocols are “shared” with the IB NBC Precert Worklist (NSR #20081010)  List Template (Main)  IB BILLING WORKLIST – EN^IBFBWL  List Template (Worklist Actions)  IB BILLING WORKLIST ACTIONS - ACTIONS^IBFBWL  List Template (Expand Authorization View)  IB BILLING WORKLIST EXPAND – EXPAND^IBFBWL  List Template (Authorization History)  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| RSD |  | | | |
| Template Type | Sort | Input | Print | Other |
| Related Options |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Routines | Description |
| --- | --- |
| DD References |  |
| Global References |  |

Table 6: Bulletins

| Bulletins | Description | | | |
| --- | --- | --- | --- | --- |
| Bulletin Name | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| RTM |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Routines | Description |
| --- | --- |
| Mail Subject |  |
| Mail Group |  |
| Parameters |  |
| DD References |  |

##### Data Entries Affected by the Design

Table 7: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| N/A |  |  |

##### Unique Record(s)

Table 8: Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
| N/A |  |  |

##### File or Global Size Changes

Table 9: File or Global Size Changes

| File/Global Name(s) | Estimated Increase | Estimated Decrease |
| --- | --- | --- |
| Integrated Billing FB Interface Tracking File (#360) | New File – Minimal per each FBCS authorization |  |

##### Mail Groups

Table 10: Mail Groups

| Mail Groups | Activities | | | |
| --- | --- | --- | --- | --- |
| Mail Group Name | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Related Options |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Mail Groups | Instructions | |
| --- | --- | --- |
| DD References |  | |
| Related Protocols |  | |
| Mail Group Description |  | |
| Self-Enrollment Allowed | Yes | No |
| Type | Public | Private |

##### Security Keys

Table : Security Keys

| Security Keys | Activities | | | |
| --- | --- | --- | --- | --- |
| Security Key Name | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Related Options |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Security Keys | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| Data Passing | Input | Output | Both | Global Reference | Local Reference |
| Security Key Description |  | | | | |
| Subordinate Keys |  | | | | |
| Mutually Exclusive Keys |  | | | | |
| Granting Condition Logic |  | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
|  |

| Security Keys | Activities |
| --- | --- |
| Hierarchical Precedence |  |

##### Options

Table : Options

| Options | Activities | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Option Name | 1. IB BILLING WORKLIST 2. IB NVC BILLING WORKLIST REPORT | | | | | | | | |
| Enhancement Category | New | | Modify | Delete | | | No Change | | |
| Associated Menu Options that will invoke this reference |  | | | | | | | | |
| Data Passing | Input | Output | | Both | | Global Reference | | | Local Reference |
| Menu Text Description |  | | | | | | | | |
| Option Type | Edit | | Print | | Menu | | | Inquire | |
| Action | | Run Routine | | Other | | |  | |
| Associated Routine | 1. EN^IBFBWL 2. EN^IBFBWLR | | | | | | | | |
| Option Definition | 1. This option will display the IB BILLING WORKLIST for Facility Revenue, RUR-SC and Billing workgroups 2. This option will display historical worklist activity by patient or date range | | | | | | | | |

| Current Entry Action Logic |
| --- |
|  |

| Modified Entry Action Logic (Changes are in bold) |
| --- |
|  |

| Current Exit Action Logic |
| --- |
|  |

| Modified Exit Action Logic (Changes are in bold) |
| --- |
|  |

##### Protocols

Table : Protocols

| Protocol Name | Activities |
| --- | --- |
|  | The Billing Worklist utilizes VistA List Manager “Lists” that are supported by “Protocols”. Some of the Worklists / Protocols are “shared” with the IB NBC Precert Worklist (NSR #20081010)  Following details the relationship between the List Templates and Protocols:  List Template (Main)  IB BILLING WORKLIST – EN^IBFBWL  Corresponding Menu (Protocol)  IB BILLING WORKLIST MENU  Action Protocols  IB BILLING WORKLIST REFRESH - REFRESH^IBFBWL  IB BILLING WORKLIST EXPAND - EXPAND^IBFBWL  IB BILLING WORKLIST CLAIMS TRACKING – LINKCT^IBFBWL  IB BILLING WORKLIST ACTIONS – ACTIONS^IBFBWL  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL  List Template (Worklist Actions)  IB BILLING WORKLIST ACTIONS - ACTIONS^IBFBWL  Corresponding Menu (Protocol)  IB BILLING WORKLIST ACTION MENU  Action Protocols  IB BILLING WORKLIST COMPLETE – DONE^IBFBWL3  IB BILLING WORKLIST REMOVE – REM^IBFBWL3  List Manager (Expand Authorization View)  IB BILLING WORKLIST EXPAND – EXPAND^IBFBWL  Corresponding Menu (Protocol)  IB BILLING WORKLIST EXPAND MENU PROTOCOL  Action Protocols  IB BILLING WORKLIST EXPAND REFRESH – REFRESH^IBFBWL2  List Manager (Authorization History)  IB BILLING WORKLIST HISTORY – HISTORY^IBFBWL  Corresponding Menu (Protocol)  None  Action Protocols  None |
| Enhancement Category | New  Modify  Delete  No Change |
| Associated Protocols |  |
| Data Passing | Input  Output  Both  Global Reference  Local Reference |
| Item Text Description |  |
| Protocol Type | Action  Menu  Protocol  Protocol Menu  Limited Protocol  Extended Action  Dialog  Other (See Above) |
| Associated Routine | ^IBWLST\* |

| Current Entry Action Logic |
| --- |
|  |

| Modified Entry Action Logic (Changes are in bold) |
| --- |
|  |

| Current Exit Action Logic |
| --- |
|  |

| Modified Exit Action Logic (Changes are in bold) |
| --- |
|  |

##### RPC

Table : RPCs

| RPCs | Activities | | |
| --- | --- | --- | --- |
| Name | N/A | | |
| TAG^RTN |  | | |
| Input Parameters |  | | |
| Results Array | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| Description |  | | |

##### Constants Defined in Interface

Table : Constants Defined in Interface

| Name | Description |
| --- | --- |
| N/A |  |

##### Variables Defined in Interface

Table : Variables Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### Types Defined in Interface

Table : Types Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### GUI

Table : GUI

| Unit Name | Description |
| --- | --- |
| N/A |  |

##### GUI Classes

Table : GUI Classes

| GUI Classes | Instructions |
| --- | --- |
| Class Name |  |
| Derived From Class |  |
| Purpose |  |

##### Current Form

##### Modified Form

##### Components on Form

Table : Components on Form

| Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### Events

Table : Events

| Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### Methods

Table : Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| N/A |  |  |

##### Special References

Table : Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### Class Events

Table : Class Events

| Name | Type | Description |
| --- | --- | --- |
| N/A |  |  |

##### Class Methods

Table : Class Methods

| Name | Procedure/Function | Description |
| --- | --- | --- |
| N/A |  |  |

##### Class Properties

Table : Class Properties

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| N/A |  |  |  |

##### Uses Clause

##### Forms

Table : Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| Form Name | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Form Functionality |  | | | |

| Current Form Layout |
| --- |
|  |

| Modified Form Layout (Changes are in bold) |
| --- |
|  |

##### Functions

Table : Functions

| Function Name | Activities | | | |
| --- | --- | --- | --- | --- |
| Short Description | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Related Options |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  |  |

| Function Name | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| DD References |  | | | | |
| Related Protocols |  | | | | |
| Related ICRs |  | | | | |
| Data Passing | Input | Output | Both | Global Reference | Local Reference |
| Input Attribute Name and Definition | Name: | | | | |
| Definition: | | | | |
| Output Attribute Name and Definition | Name: | | | | |
| Definition: | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
|  |

##### Dialog

Table : Dialog

| Dialog | Instructions | | | |
| --- | --- | --- | --- | --- |
| Dialog Message (Description) | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Dialog Message (Description) Condition |  | | | |
| Current Dialog Message (Description) |  | | | |
| Modified Dialog Message (Description)  (Changes are in bold) |  | | | |

##### Help Frame

Table : Help Frame

| Help Frame | Description | | | |
| --- | --- | --- | --- | --- |
| Help Frame Text | N/A | | | |
| Enhancement Category | New | Modify | Delete | No Change |
| Help Frame Text Calling Mechanism |  | | | |

| Current Help Frame Text |
| --- |
|  |

| Modified Help Frame Text (Changes are in bold) |
| --- |
|  |

##### HL7 Application Parameter

Table : HL7 Application Parameter

| HL7 Application Parameter Name | Description |
| --- | --- |

| Enhancement Category | New | Modify | Delete | No Change |
| --- | --- | --- | --- | --- |

| Application Status | Active | Inactive | Active | Inactive |
| --- | --- | --- | --- | --- |

| Enhancement Category | Current | Modified |
| --- | --- | --- |
| Facility Name | N/A |  |
| Country Code |  |  |
| HL7 Field Separator |  |  |
| HL7 Encoding Characters |  |  |
| Mail Group |  |  |

##### HL7 Logical Link

Table : HL7 Logical Link

| HL7 Logical Link | Description |
| --- | --- |
| HL7 Logical Link Parameter Name | N/A |

| Enhancement Category | New | Modify | Delete | No Change |
| --- | --- | --- | --- | --- |

| Enhancement Category | Current | Modified |
| --- | --- | --- |
| Node |  |  |
| Institution |  |  |
| Domain |  |  |
| Autostart |  |  |
| Queue Size |  |  |
| LLP Type |  |  |

##### COTS Interface

Table : COTS Interface

| COTS Interface | Description |
| --- | --- |
| Communication Method | N/A |
| Application Interface |  |

## Network Detailed Design

There are no network topology changes related to this project.

## Security and Privacy

This project does not add any additional security or privacy design considerations.

### Security

Security and Privacy are controlled by the VistA application and the underlying operating system. This application does not mandate any additional security or privacy requirements.

### Privacy

Security and Privacy are controlled by the VistA application and the underlying operating system. This application does not mandate any additional security or privacy requirements.

## SOA/ESS Detailed Design

Not applicable – reference Section 4.4.

# External System Interface Design

Not applicable – there is no external system involved with this project.

# Human-Machine Interface

Not applicable – this project does not change the human-machine interface, which is done via the VistA user options.

# Attachment A – Approval Signatures

The signature below is an acknowledgement that the signatory understands the purpose and content of this document. By signing this document, you agree to this as the SDD to begin work on the project described within, and commitment of the necessary resources.

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

xxxxx

Office of Information & Technology (OI&T) Project Manager (PjM), Date

Integrated Project Team (IPT) Co-Chair

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

xxxx

Business Sponsor Date

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

xxxx

VHA PjM, IPT Co-Chair Date

# Attachment B – Acronyms

Table : Acronym List

| Term | Definition |
| --- | --- |
| AITC | Austin Information Technology Center |
| API | Application Programming Interface |
| BN | Business Need |
| BRD | Business Requirements Document |
| CBO | Chief Business Office |
| CBOC | Community Based Outpatient Clinic |
| CLC | Community Living Center |
| COTS | Commercial Off-the-Shelf |
| CP&E | Claims Processing and Eligibility |
| CPRS | Computerized Patient Record System |
| DBMS | Database Management System |
| DD | Data Dictionary |
| EDI | Electronic Data Interchange |
| EHR | Electronic Health Record |
| ERD | Entity Relationship Diagram |
| ERR | Enterprise Requirements Repository |
| ESS | Enterprise Shared Services |
| FB | Fee Basis |
| FBCS | Fee Basis Claims System |
| GUI | Graphical User Interface |
| HL7 | Health Level 7 |
| IB | Integrated Billing |
| ICR | Integration Control Registration |
| IPT | Integrated Project Team |
| LAN | Local Area Network |
| MCCF | Medical Care Collection Fund |
| MCCR | Medical Care Cost Recovery |
| MDWS | Medical Domain Web Services |
| MUMPS | Massachusetts General Hospital Utility Multi-Programming System |
| NFPO | National Fee Program Office |
| OI&T | Office of Information & Technology |
| OIG | Office of Inspector General |
| OWNR | Owner Requirement |
| PjM | Project Manager |
| RPC | Remote Procedure Call |
| RSD | Requirements Specification Document |
| RTM | Requirements Traceability Matrix |
| RUR | Revenue Utilization Review |
| SDD | System Design Document |
| SOA | Service Oriented Architecture |
| TSPR | Technical Services Project Repository |
| VA | Department of Veterans Affairs |
| VACO | VA Central Office |
| VAMC | VA Medical Center |
| VHA | Veterans Health Administration |
| VISN | Veterans Integrated Service Networks |
| VistA | Veterans Health Information Systems and Technology Architecture |

# Attachment C – Additional Information

This section and its sub-sections are to be determined. Blank sections have been kept for version control.

## A.1 Identification of Technology and Standards

## A.2 Constraining Policies, Directives and Procedures

## A.3 RTM

The Fee-Revenue Enhancements RTM is available via TSPR here: [http:/xxxxxxxxxxxxxxxxx/warboard/anotebk.asp?proj=1786&Type=Active](http://xxxxxxxxxxxxxxxxx/warboard/anotebk.asp?proj=1786&Type=Active)

Until the RTM is approved and posted to TSPE, it will be made available upon request from the project team.

## A.4 Packaging and Installation

## A.5 Design Metrics

# Attachment D – Change Log

| Change Log Version | Change | Notes | Status/Resolution |
| --- | --- | --- | --- |
| 0.02 | Globally: Verbiage updated to match across all 5 SDDs |  |  |
| 0.02 | Globally: Ensure acronyms are defined at first mention; update Acronyms table accordingly |  |  |
| 0.02 | Globally: 508 diagrams – add alternative text |  |  |
| 0.02 | Sections 3.1.1 and 3.2.1: Other SDDs have figure here; need to include one? |  |  |
| 0.02 | Section 8: Was removed in latest version; replaced here for outline consistency |  |  |

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| June 2015 | 2.10 | Changed Heading 1 default setting to eliminate page break before | Process Management |
| May 2015 | 2.9 | Edited for Section 508 conformance and remediated with Common Look Office tool | Process Management |
| February 2015 | 2.8 | Incorporates revisions from PMAS Reform Lockdown; namely removing requirements for information that can be obtained from other PMAS authoritative sources. | Andrew Slawter, Office of Technology Strategies |
| 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 |

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>