Department of Veterans Affairs

**Compensation and Pension Records**

**Interchange (CAPRI)**

System Design Document

Patch DVBA\*2.7\*193



October 2016

Version 1.2

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 1/26/2016 | 1.0 | Created | CSRA Team |
| 10/10/2016 | 1.1 | Updated | CSRA Team |
| 10/18/2016 | 1.2 | Updated based on AERB review. | CSRA Team |

Artifact Rationale

The System Design Document (SDD) is a dual-use document that provides the conceptual design as well as the as-built design. This document will be updated as the product is built, to reflect the as-built product.

When to Complete Each Section of the SDD

| Section | Completed On or Before PMAS Phase | Rationale |
| --- | --- | --- |
| 1 – Introduction | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 2 - Background | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 3 – Conceptual Design | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 4 – System Architecture | MS 0 Review; updated thereafter | Conceptual design should inform evaluation of investments |
| 5 – Data Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 6 – Detailed Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 7 – External System Interface Design | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| 8 – Human Machine Interfaces | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |
| Attachments | MS 1 Review; updated thereafter | Design details should be elaborated upon during PMAS Planning phase and prior to development |

A product’s system design should be defined conceptually prior to the allocation of personnel and resources that occur at project initiation. This gives the enterprise an opportunity to evaluate IT investments before project teams are stood up and funding is allocated. Sections 1- 4 which discuss the high level design should be completed prior to MS 0. All sections should be completed and updated before MS 1. Projects will need to address all SDD approval constraints prior to the MS 2 review. In addition, the SDD should reflect the as-built product going into the MS 2 review.

Table of Contents

[1. Introduction 1](#_Toc464574621)

[1.1. Scope 1](#_Toc464574622)

[1.2. User Profiles 1](#_Toc464574623)

[2. Background 1](#_Toc464574624)

[2.1. Overview of the System 1](#_Toc464574625)

[2.2. Overview of the Business Process 2](#_Toc464574626)

[2.3. Overview of the Significant Requirements 3](#_Toc464574627)

[2.3.1. Overview of Significant Functional Requirements 3](#_Toc464574628)

[2.3.2. Overview of Functional Workload / Performance Requirements 4](#_Toc464574629)

[2.3.3. Overview of Operational Requirements 4](#_Toc464574630)

[2.3.4. Overview of the Technical Requirements 5](#_Toc464574631)

[2.3.5. Overview of the Security or Privacy Requirements 5](#_Toc464574632)

[2.3.6. Overview of System Criticality and High Availability Requirements 5](#_Toc464574633)

[2.3.7. Single Sign-on Requirement 5](#_Toc464574634)

[2.3.8. Requirement for Use of Enterprise Portals 5](#_Toc464574635)

[2.3.9. Special Device Requirements 5](#_Toc464574636)

[3. Conceptual Design 6](#_Toc464574637)

[3.1. Conceptual Application Design 6](#_Toc464574638)

[3.1.1. Application Context 6](#_Toc464574639)

[3.1.2. High-Level Application Design 11](#_Toc464574640)

[3.1.3. Application Locations 11](#_Toc464574641)

[3.2. Conceptual Data Design 12](#_Toc464574642)

[3.2.1. Project Conceptual Data Model 12](#_Toc464574643)

[3.2.2. Database Information 12](#_Toc464574644)

[3.2.3. User Interface Data Mapping 12](#_Toc464574645)

[3.2.3.1. Application Screen Interface 12](#_Toc464574646)

[3.2.3.2. Application Report Interface 12](#_Toc464574647)

[3.2.3.3. Unmapped Data Element 13](#_Toc464574648)

[3.3. Conceptual Infrastructure Design 13](#_Toc464574649)

[3.3.1. System Criticality and High Availability 13](#_Toc464574650)

[3.3.2. Special Technology 13](#_Toc464574651)

[3.3.3. Technology Locations 13](#_Toc464574652)

[3.3.4. Conceptual Infrastructure Diagram 13](#_Toc464574653)

[4. System Architecture 13](#_Toc464574654)

[4.1. Hardware Architecture 13](#_Toc464574655)

[4.2. Software Architecture 15](#_Toc464574656)

[4.3. Network Architecture 16](#_Toc464574657)

[4.4. Service Oriented Architecture / ESS 16](#_Toc464574658)

[4.4.1. Services Provided 16](#_Toc464574659)

[4.4.1.1. DAS 16](#_Toc464574660)

[4.4.1.2. Virtual VA 17](#_Toc464574661)

[CAPRI sends patient electronic reports for storage in Virtual VA. 17](#_Toc464574662)

[4.4.2. Services Required / Consumed 17](#_Toc464574663)

[4.4.2.1. DAS 17](#_Toc464574664)

[4.4.2.2. Virtual VA 17](#_Toc464574665)

[4.4.2.3. HealthEVet Web Service 17](#_Toc464574666)

[4.5. Enterprise Architecture 17](#_Toc464574667)

[5. Data Design 20](#_Toc464574668)

[5.1. DBMS Files 20](#_Toc464574669)

[5.2. Non-DBMS Files 20](#_Toc464574670)

[5.3. Data View 20](#_Toc464574671)

[6. Detailed Design 21](#_Toc464574672)

[6.1. Hardware Detailed Design 21](#_Toc464574673)

[6.2. Software Detailed Design 21](#_Toc464574674)

[6.2.1. Conceptual Design 21](#_Toc464574675)

[6.2.2. Specific Requirements 21](#_Toc464574676)

[6.2.2.1. Requirement 2.6.1 21](#_Toc464574677)

[6.2.2.2. Database Repository 22](#_Toc464574678)

[6.2.2.3. System Features 22](#_Toc464574679)

[6.2.2.4. Design Element Tables 26](#_Toc464574680)

[6.2.2.4.1. Routines (Entry Points) 26](#_Toc464574681)

[6.2.2.4.2. Templates – N/A for Requirement 2.6.1 96](#_Toc464574682)

[6.2.2.4.3. Bulletins 96](#_Toc464574683)

[6.2.2.4.4. Unique Record(s) – N/A for Requirement 2.6.1 97](#_Toc464574684)

[6.2.2.4.5. File or Global Size Changes – N/A for Requirement 2.6.1 97](#_Toc464574685)

[6.2.2.4.6. Mail Groups 97](#_Toc464574686)

[6.2.2.4.7. Security Keys 97](#_Toc464574687)

[6.2.2.4.8. Options 98](#_Toc464574688)

[6.2.2.4.9. Protocols – N/A for Requirement 2.6.1 99](#_Toc464574689)

[6.2.2.4.10. Remote Procedure Call (RPC) 99](#_Toc464574690)

[6.2.2.4.11. Constants Defined in Interface – N/A for Requirement 2.6.1 102](#_Toc464574691)

[6.2.2.4.12. Variables Defined in Interface – N/A for Requirement 2.6.1 102](#_Toc464574692)

[6.2.2.4.13. Types Defined in Interface – N/A for Requirement 2.6.1 102](#_Toc464574693)

[6.2.2.4.14. GUI – ViewExam.pas 102](#_Toc464574694)

[6.2.2.4.15. GUI Classes – N/A for Requirement 2.6.1 103](#_Toc464574695)

[6.2.2.4.16. Current Form 103](#_Toc464574696)

[6.2.2.4.17. Modified Form 105](#_Toc464574697)

[6.2.2.4.18. Components on Form 106](#_Toc464574698)

[6.2.2.4.19. Events 107](#_Toc464574699)

[6.2.2.4.20. Methods 108](#_Toc464574700)

[6.2.2.4.21. Special References 108](#_Toc464574701)

[6.2.2.4.22. Class Events – N/A for Requirement 2.6.1 109](#_Toc464574702)

[6.2.2.4.23. Class Methods – N/A for Requirement 2.6.1 109](#_Toc464574703)

[6.2.2.4.24. Class Properties – N/A for Requirement 2.6.1 109](#_Toc464574704)

[6.2.2.4.25. Uses Clause – N/A for Requirement 2.6.1 109](#_Toc464574705)

[6.2.2.4.26. Forms 109](#_Toc464574706)

[6.2.2.4.27. Functions 109](#_Toc464574707)

[6.2.2.4.28. GUI – Main.pas 136](#_Toc464574708)

[6.2.2.4.29. GUI Classes – N/A for Requirement 2.6.1 137](#_Toc464574709)

[6.2.2.4.30. Current Form 137](#_Toc464574710)

[6.2.2.4.31. Modified Form 137](#_Toc464574711)

[6.2.2.4.32. Components on Form 137](#_Toc464574712)

[6.2.2.4.33. Events 137](#_Toc464574713)

[6.2.2.4.34. Methods – N/A for Requirement 2.6.1 138](#_Toc464574714)

[6.2.2.4.35. Special References – N/A for Requirement 2.6.1 138](#_Toc464574715)

[6.2.2.4.36. Class Events – N/A for Requirement 2.6.1 138](#_Toc464574716)

[6.2.2.4.37. Class Methods – N/A for Requirement 2.6.1 138](#_Toc464574717)

[6.2.2.4.38. Class Properties – N/A for Requirement 2.6.1 138](#_Toc464574718)

[6.2.2.4.39. Uses Clause 138](#_Toc464574719)

[6.2.2.4.40. Forms 138](#_Toc464574720)

[6.2.2.4.41. Functions 138](#_Toc464574721)

[6.2.2.4.42. GUI – Reports.pas 204](#_Toc464574722)

[6.2.2.4.43. GUI Classes – N/A in Requirement 2.6.1 205](#_Toc464574723)

[6.2.2.4.44. Current Form 205](#_Toc464574724)

[6.2.2.4.45. Modified Form 207](#_Toc464574725)

[6.2.2.4.46. Components on Form – N/A for Requirement 2.6.1 208](#_Toc464574726)

[6.2.2.4.47. Events 208](#_Toc464574727)

[6.2.2.4.48. Methods 208](#_Toc464574728)

[6.2.2.4.49. Special References – N/A in Requirement 2.6.1 209](#_Toc464574729)

[6.2.2.4.50. Class Events – N/A in Requirement 2.6.1 209](#_Toc464574730)

[6.2.2.4.51. Class Methods – N/A in Requirement 2.6.1 209](#_Toc464574731)

[6.2.2.4.52. Class Properties – N/A in Requirement 2.6.1 209](#_Toc464574732)

[6.2.2.4.53. Uses Clause – N/A in Requirement 2.6.1 209](#_Toc464574733)

[6.2.2.4.54. Forms 209](#_Toc464574734)

[6.2.2.4.55. Functions 209](#_Toc464574735)

[6.2.2.4.56. GUI – ReRouteReject.pas 219](#_Toc464574736)

[6.2.2.4.57. GUI Classes 219](#_Toc464574737)

[6.2.2.4.58. Current Form 219](#_Toc464574738)

[6.2.2.4.59. Modified Form 220](#_Toc464574739)

[6.2.2.4.60. Components on Form 220](#_Toc464574740)

[6.2.2.4.61. Events 220](#_Toc464574741)

[6.2.2.4.62. Methods 220](#_Toc464574742)

[6.2.2.4.63. Special References 220](#_Toc464574743)

[6.2.2.4.64. Class Events – N/A for Requirement 2.6.1 221](#_Toc464574744)

[6.2.2.4.65. Class Methods – N/A for Requirement 2.6.1 221](#_Toc464574745)

[6.2.2.4.66. Class Properties – N/A for Requirement 2.6.1 221](#_Toc464574746)

[6.2.2.4.67. Uses Clause – N/A for Requirement 2.6.1 221](#_Toc464574747)

[6.2.2.4.68. Forms 221](#_Toc464574748)

[6.2.2.4.69. Functions 221](#_Toc464574749)

[6.2.2.4.70. GUI – edtRerouteReasons.pas 226](#_Toc464574750)

[6.2.2.4.71. GUI Classes 226](#_Toc464574751)

[6.2.2.4.72. Current Form 226](#_Toc464574752)

[6.2.2.4.73. Modified Form 227](#_Toc464574753)

[6.2.2.4.74. Components on Form 227](#_Toc464574754)

[6.2.2.4.75. Events 228](#_Toc464574755)

[6.2.2.4.76. Methods 229](#_Toc464574756)

[6.2.2.4.77. Special References 229](#_Toc464574757)

[6.2.2.4.78. Class Events – N/A in Requirement 2.6.1 229](#_Toc464574758)

[6.2.2.4.79. Class Methods – N/A in Requirement 2.6.1 229](#_Toc464574759)

[6.2.2.4.80. Class Properties – N/A in Requirement 2.6.1 229](#_Toc464574760)

[6.2.2.4.81. Uses Clause 229](#_Toc464574761)

[6.2.2.4.82. Forms 229](#_Toc464574762)

[6.2.2.4.83. Functions 230](#_Toc464574763)

[6.2.2.4.84. GUI – ReRouteRequest.pas 255](#_Toc464574764)

[6.2.2.4.85. GUI Classes – N/A for Requirement 2.6.1 255](#_Toc464574765)

[6.2.2.4.86. Current Form 255](#_Toc464574766)

[6.2.2.4.87. Modified Form 255](#_Toc464574767)

[6.2.2.4.88. Components on Form 255](#_Toc464574768)

[6.2.2.4.89. Events 256](#_Toc464574769)

[6.2.2.4.90. Methods 257](#_Toc464574770)

[6.2.2.4.91. Special References 258](#_Toc464574771)

[6.2.2.4.92. Class Events – Not Applicable for Requirement 2.6.1 258](#_Toc464574772)

[6.2.2.4.93. Class Methods – Not Applicable for Requirement 2.6.1 258](#_Toc464574773)

[6.2.2.4.94. Class Properties – Not Applicable for Requirement 2.6.1 258](#_Toc464574774)

[6.2.2.4.95. Uses Clause 258](#_Toc464574775)

[6.2.2.4.96. Forms 258](#_Toc464574776)

[6.2.2.4.97. Functions 259](#_Toc464574777)

[6.2.2.4.98. GUI – DemTranMain.pas 290](#_Toc464574778)

[6.2.2.4.99. GUI Classes – N/A for Requirement 2.6.1 291](#_Toc464574779)

[6.2.2.4.100. Current Form – N/A for Requirement 2.6.1 291](#_Toc464574780)

[6.2.2.4.101. Modified Form – N/A for Requirement 2.6.1 291](#_Toc464574781)

[6.2.2.4.102. Components on Form – N/A for Requirement 2.6.1 291](#_Toc464574782)

[6.2.2.4.103. Events – N/A for Requirement 2.6.1 291](#_Toc464574783)

[6.2.2.4.104. Methods 291](#_Toc464574784)

[6.2.2.4.105. Special References – N/A for Requirement 2.6.1 291](#_Toc464574785)

[6.2.2.4.106. Class Events – N/A for Requirement 2.6.1 291](#_Toc464574786)

[6.2.2.4.107. Class Methods – N/A for Requirement 2.6.1 291](#_Toc464574787)

[6.2.2.4.108. Class Properties – N/A for Requirement 2.6.1 291](#_Toc464574788)

[6.2.2.4.109. Uses Clause 291](#_Toc464574789)

[6.2.2.4.110. Forms – N/A for Requirement 2.6.1 291](#_Toc464574790)

[6.2.2.4.111. Functions 291](#_Toc464574791)

[6.2.2.4.112. GUI – frmContractedExamCheckIn.pas 295](#_Toc464574792)

[6.2.2.4.113. GUI Classes – N/A for Requirement 2.6.1 296](#_Toc464574793)

[6.2.2.4.114. Current Form – N/A for Requirement 2.6.1 296](#_Toc464574794)

[6.2.2.4.115. Modified Form – N/A for Requirement 2.6.1 296](#_Toc464574795)

[6.2.2.4.116. Components on Form – N/A for Requirement 2.6.1 296](#_Toc464574796)

[6.2.2.4.117. Events – N/A for Requirement 2.6.1 296](#_Toc464574797)

[6.2.2.4.118. Methods 296](#_Toc464574798)

[6.2.2.4.119. Special References – N/A for Requirement 2.6.1 296](#_Toc464574799)

[6.2.2.4.120. Class Events – N/A for Requirement 2.6.1 296](#_Toc464574800)

[6.2.2.4.121. Class Methods – N/A for Requirement 2.6.1 296](#_Toc464574801)

[6.2.2.4.122. Class Properties – N/A for Requirement 2.6.1 296](#_Toc464574802)

[6.2.2.4.123. Uses Clause 296](#_Toc464574803)

[6.2.2.4.124. Forms – N/A for Requirement 2.6.1 296](#_Toc464574804)

[6.2.2.4.125. Functions 296](#_Toc464574805)

[6.2.2.4.126. GUI – frmContractedExamDeleteCO.pas 299](#_Toc464574806)

[6.2.2.4.127. GUI Classes – N/A for Requirement 2.6.1 299](#_Toc464574807)

[6.2.2.4.128. Current Form – N/A for Requirement 2.6.1 299](#_Toc464574808)

[6.2.2.4.129. Modified Form – N/A for Requirement 2.6.1 299](#_Toc464574809)

[6.2.2.4.130. Components on Form – N/A for Requirement 2.6.1 299](#_Toc464574810)

[6.2.2.4.131. Events – N/A for Requirement 2.6.1 299](#_Toc464574811)

[6.2.2.4.132. Methods 299](#_Toc464574812)

[6.2.2.4.133. Special References – N/A for Requirement 2.6.1 300](#_Toc464574813)

[6.2.2.4.134. Class Events – N/A for Requirement 2.6.1 300](#_Toc464574814)

[6.2.2.4.135. Class Methods – N/A for Requirement 2.6.1 300](#_Toc464574815)

[6.2.2.4.136. Class Properties – N/A for Requirement 2.6.1 300](#_Toc464574816)

[6.2.2.4.137. Uses Clause 300](#_Toc464574817)

[6.2.2.4.138. Forms – N/A for Requirement 2.6.1 300](#_Toc464574818)

[6.2.2.4.139. Functions 300](#_Toc464574819)

[6.2.2.4.140. GUI – ManageReportsCCR.pas 302](#_Toc464574820)

[6.2.2.4.141. GUI Classes – N/A for Requirement 2.6.1 302](#_Toc464574821)

[6.2.2.4.142. Current Form – N/A for Requirement 2.6.1 302](#_Toc464574822)

[6.2.2.4.143. Modified Form – N/A for Requirement 2.6.1 302](#_Toc464574823)

[6.2.2.4.144. Components on Form – N/A for Requirement 2.6.1 302](#_Toc464574824)

[6.2.2.4.145. Events 302](#_Toc464574825)

[6.2.2.4.146. Methods – N/A for Requirement 2.6.1 303](#_Toc464574826)

[6.2.2.4.147. Special References – N/A for Requirement 2.6.1 303](#_Toc464574827)

[6.2.2.4.148. Class Events – N/A for Requirement 2.6.1 303](#_Toc464574828)

[6.2.2.4.149. Class Methods – N/A for Requirement 2.6.1 303](#_Toc464574829)

[6.2.2.4.150. Class Properties – N/A for Requirement 2.6.1 303](#_Toc464574830)

[6.2.2.4.151. Uses Clause – N/A for Requirement 2.6.1 303](#_Toc464574831)

[6.2.2.4.152. Forms – N/A for Requirement 2.6.1 303](#_Toc464574832)

[6.2.2.4.153. Functions 303](#_Toc464574833)

[6.2.2.4.154. GUI – ManageReports.pas 310](#_Toc464574834)

[6.2.2.4.155. GUI Classes – N/A for Requirement 2.6.1 310](#_Toc464574835)

[6.2.2.4.156. Current Form – N/A for Requirement 2.6.1 310](#_Toc464574836)

[6.2.2.4.157. Modified Form – N/A for Requirement 2.6.1 310](#_Toc464574837)

[6.2.2.4.158. Components on Form – N/A for Requirement 2.6.1 310](#_Toc464574838)

[6.2.2.4.159. Events 310](#_Toc464574839)

[6.2.2.4.160. Methods – N/A for Requirement 2.6.1 310](#_Toc464574840)

[6.2.2.4.161. Special References – N/A for Requirement 2.6.1 310](#_Toc464574841)

[6.2.2.4.162. Class Events – N/A for Requirement 2.6.1 310](#_Toc464574842)

[6.2.2.4.163. Class Methods – N/A for Requirement 2.6.1 310](#_Toc464574843)

[6.2.2.4.164. Class Properties – N/A for Requirement 2.6.1 310](#_Toc464574844)

[6.2.2.4.165. Uses Clause – N/A for Requirement 2.6.1 310](#_Toc464574845)

[6.2.2.4.166. Forms – N/A for Requirement 2.6.1 310](#_Toc464574846)

[6.2.2.4.167. Functions 310](#_Toc464574847)

[6.2.2.4.168. GUI – TIUSign.pas 317](#_Toc464574848)

[6.2.2.4.169. GUI Classes – N/A for Requirement 2.6.1 317](#_Toc464574849)

[6.2.2.4.170. Current Form – N/A for Requirement 2.6.1 317](#_Toc464574850)

[6.2.2.4.171. Modified Form – N/A for Requirement 2.6.1 317](#_Toc464574851)

[6.2.2.4.172. Components on Form – N/A for Requirement 2.6.1 317](#_Toc464574852)

[6.2.2.4.173. Events – N/A for Requirement 2.6.1 317](#_Toc464574853)

[6.2.2.4.174. Methods 317](#_Toc464574854)

[6.2.2.4.175. Special References – N/A for Requirement 2.6.1 317](#_Toc464574855)

[6.2.2.4.176. Class Events – N/A for Requirement 2.6.1 317](#_Toc464574856)

[6.2.2.4.177. Class Methods – N/A for Requirement 2.6.1 317](#_Toc464574857)

[6.2.2.4.178. Class Properties – N/A for Requirement 2.6.1 317](#_Toc464574858)

[6.2.2.4.179. Uses Clause – N/A for Requirement 2.6.1 317](#_Toc464574859)

[6.2.2.4.180. Forms – N/A for Requirement 2.6.1 317](#_Toc464574860)

[6.2.2.4.181. Functions 317](#_Toc464574861)

[6.2.2.4.182. GUI – UncosignedUtility.pas 322](#_Toc464574862)

[6.2.2.4.183. GUI Classes – N/A for Requirement 2.6.1 322](#_Toc464574863)

[6.2.2.4.184. Current Form – N/A for Requirement 2.6.1 322](#_Toc464574864)

[6.2.2.4.185. Modified Form – N/A for Requirement 2.6.1 322](#_Toc464574865)

[6.2.2.4.186. Components on Form – N/A for Requirement 2.6.1 322](#_Toc464574866)

[6.2.2.4.187. Events 322](#_Toc464574867)

[6.2.2.4.188. Methods – N/A for Requirement 2.6.1 323](#_Toc464574868)

[6.2.2.4.189. Special References – N/A for Requirement 2.6.1 323](#_Toc464574869)

[6.2.2.4.190. Class Events – N/A for Requirement 2.6.1 323](#_Toc464574870)

[6.2.2.4.191. Class Methods – N/A for Requirement 2.6.1 323](#_Toc464574871)

[6.2.2.4.192. Class Properties – N/A for Requirement 2.6.1 323](#_Toc464574872)

[6.2.2.4.193. Uses Clause – N/A for Requirement 2.6.1 323](#_Toc464574873)

[6.2.2.4.194. Forms – N/A for Requirement 2.6.1 323](#_Toc464574874)

[6.2.2.4.195. Functions 323](#_Toc464574875)

[6.2.2.4.196. Dialog – Not Applicable for Requirement 2.6.1 330](#_Toc464574876)

[6.2.2.4.197. Help Frame – Not Applicable for Requirement 2.6.1 330](#_Toc464574877)

[6.2.2.4.198. HL7 Application Parameter – Not Applicable for Requirement 2.6.1 330](#_Toc464574878)

[6.2.2.4.199. HL7 Logical Link – Not Applicable for Requirement 2.6.1 330](#_Toc464574879)

[6.2.2.4.200. COTS Interface – Not Applicable for Requirement 2.6.1 330](#_Toc464574880)

[6.2.3.1. Requirement 2.6.2 330](#_Toc464574881)

[6.2.3.2. Database Repository – N/A for Requirement 2.6.2. 330](#_Toc464574882)

[6.2.3.3. System Features 330](#_Toc464574883)

[6.2.3.4. Design Element Tables 331](#_Toc464574884)

[6.2.3.4.1. Routines (Entry Points) – N/A for Requirement 2.6.2 331](#_Toc464574885)

[6.2.3.4.2. Templates – N/A for Requirement 2.6.2 331](#_Toc464574886)

[6.2.3.4.3. Bulletins – N/A for Requirement 2.6.2 331](#_Toc464574887)

[6.2.3.4.4. Data Entries Affected by the Design – N/A for Requirement 2.6.2 331](#_Toc464574888)

[6.2.3.4.5. Unique Record(s) - N/A for Requirement 2.6.2 331](#_Toc464574889)

[6.2.3.4.6. File or Global Size Changes – N/A for Requirement 2.6.2 331](#_Toc464574890)

[6.2.3.4.7. Mail Groups – N/A for Requirement 2.6.2 331](#_Toc464574891)

[6.2.3.4.8. Security Keys – N/A for Requirement 2.6.2 331](#_Toc464574892)

[6.2.3.4.9. Options – N/A for Requirement 2.6.2 331](#_Toc464574893)

[6.2.3.4.10. Protocols – N/A for Requirement 2.6.2 331](#_Toc464574894)

[6.2.3.4.11. Remote Procedure Call (RPC) – N/A for Requirement 2.6.2 331](#_Toc464574895)

[6.2.3.4.12. Constants Defined in Interface – N/A for Requirement 2.6.2 331](#_Toc464574896)

[6.2.3.4.13. Variables Defined in Interface – N/A for Requirement 2.6.2 331](#_Toc464574897)

[6.2.3.4.14. Types Defined in Interface – N/A for Requirement 2.6.2 331](#_Toc464574898)

[6.2.3.4.15. GUI – Main.pas 331](#_Toc464574899)

[6.2.3.4.16. GUI Classes – N/A for Requirement 2.6.3 331](#_Toc464574900)

[6.2.3.4.17. Current Form – N/A for Requirement 2.6.2 331](#_Toc464574901)

[6.2.3.4.18. Modified Form – N/A for Requirement 2.6.2 331](#_Toc464574902)

[6.2.3.4.19. Components on Form – N/A for Requirement 2.6.2 331](#_Toc464574903)

[6.2.3.4.20. Events 331](#_Toc464574904)

[6.2.3.4.21. Methods 332](#_Toc464574905)

[6.2.3.4.22. Special References – N/A for Requirement 2.6.3 332](#_Toc464574906)

[6.2.3.4.23. Class Events – N/A for Requirement 2.6.3 332](#_Toc464574907)

[6.2.3.4.24. Class Methods – N/A for Requirement 2.6.2 332](#_Toc464574908)

[6.2.3.4.25. Class Properties – N/A for Requirement 2.6.2 332](#_Toc464574909)

[6.2.3.4.26. Uses Clause – N/A for Requirement 2.6.2 332](#_Toc464574910)

[6.2.3.4.27. Forms – N/A for Requirement 2.6.2 332](#_Toc464574911)

[6.2.3.4.28. Functions 332](#_Toc464574912)

[6.2.3.4.29. Dialog – N/A for Requirement 2.6.2 342](#_Toc464574913)

[6.2.3.4.30. Help Frame – N/A for Requirement 2.6.2 342](#_Toc464574914)

[6.2.3.4.31. HL7 Application Parameter – N/A for Requirement 2.6.2 342](#_Toc464574915)

[6.2.3.4.32. HL7 Logical Link – N/A for Requirement 2.6.2 342](#_Toc464574916)

[6.2.3.4.33. COTS Interface – N/A for Requirement 2.6.2 342](#_Toc464574917)

[6.2.4.1. Requirement 2.6.3 342](#_Toc464574918)

[CAPRI shall be modified during the BHIE transition to have the DoD tab disabled within CAPRI. The business will notify the development team when the disabling should occur. (Business Requirements 608696)Database Repository – N/A for Requirement 2.6.3 342](#_Toc464574919)

[6.2.4.2. Database Repository – N/A for Requirement 2.6.3. 342](#_Toc464574920)

[6.2.4.3. System Features 342](#_Toc464574921)

[6.2.4.4. Design Element Tables 343](#_Toc464574922)

[6.2.4.4.1. Routines (Entry Points) – N/A for Requirement 2.6.2 343](#_Toc464574923)

[6.2.4.4.2. Templates – N/A for Requirement 2.6.2 343](#_Toc464574924)

[6.2.4.4.3. Bulletins – N/A for Requirement 2.6.2 343](#_Toc464574925)

[6.2.4.4.4. Data Entries Affected by the Design – N/A for Requirement 2.6.2 343](#_Toc464574926)

[6.2.4.4.5. Unique Record(s) – N/A for Requirement 2.6.2 343](#_Toc464574927)

[6.2.4.4.6. File or Global Size Changes – N/A for Requirement 2.6.2 343](#_Toc464574928)

[6.2.4.4.7. Mail Groups – N/A for Requirement 2.6.2 343](#_Toc464574929)

[6.2.4.4.8. Security Keys – N/A for Requirement 2.6.2 343](#_Toc464574930)

[6.2.4.4.9. Options – N/A for Requirement 2.6.2 343](#_Toc464574931)

[6.2.4.4.10. Protocols – N/A for Requirement 2.6.2 343](#_Toc464574932)

[6.2.4.4.11. Remote Procedure Call (RPC) – N/A for Requirement 2.6.2 343](#_Toc464574933)

[6.2.4.4.12. Constants Defined in Interface – N/A for Requirement 2.6.2 343](#_Toc464574934)

[6.2.4.4.13. Variables Defined in Interface – N/A for Requirement 2.6.2 343](#_Toc464574935)

[6.2.4.4.14. Types Defined in Interface – N/A for Requirement 2.6.2 343](#_Toc464574936)

[6.2.4.4.15. GUI – PatientList.pas 343](#_Toc464574937)

[6.2.4.4.16. GUI Classes – N/A for Requirement 2.6.2 343](#_Toc464574938)

[6.2.4.4.17. Current Form - N/A for Requirement 2.6.2 343](#_Toc464574939)

[6.2.4.4.18. Modified Form – N/A for Requirement 2.6.2 343](#_Toc464574940)

[6.2.4.4.19. Components on Form – N/A for Requirement 2.6.2 343](#_Toc464574941)

[6.2.4.4.20. Events 343](#_Toc464574942)

[6.2.4.4.21. Methods 344](#_Toc464574943)

[6.2.4.4.22. Special References – N/A for Requirement 2.6.2 344](#_Toc464574944)

[6.2.4.4.23. Class Events – N/A for Requirement 2.6.2 344](#_Toc464574945)

[6.2.4.4.24. Class Methods – N/A for Requirement 2.6.2 344](#_Toc464574946)

[6.2.4.4.25. Class Properties – N/A for Requirement 2.6.2 344](#_Toc464574947)

[6.2.4.4.26. Uses Clause – N/A for Requirement 2.6.2 344](#_Toc464574948)

[6.2.4.4.27. Forms – N/A for Requirement 2.6.2 344](#_Toc464574949)

[6.2.4.4.28. Functions 344](#_Toc464574950)

[6.2.4.4.29. Dialog – N/A for Requirement 2.6.2 368](#_Toc464574951)

[6.2.4.4.30. GUI – Main.pas 368](#_Toc464574952)

[6.2.4.4.31. GUI Classes – N/A for Requirement 2.6.2 368](#_Toc464574953)

[6.2.4.4.32. Current Form – N/A for Requirement 2.6.2 368](#_Toc464574954)

[6.2.4.4.33. Modified Form – N/A for Requirement 2.6.2 368](#_Toc464574955)

[6.2.4.4.34. Components on Form – N/A for Requirement 2.6.2 368](#_Toc464574956)

[6.2.4.4.35. Events 368](#_Toc464574957)

[6.2.4.4.36. Methods 369](#_Toc464574958)

[6.2.4.4.37. Special References – N/A for Requirement 2.6.2 369](#_Toc464574959)

[6.2.4.4.38. Class Events – N/A for Requirement 2.6.2 369](#_Toc464574960)

[6.2.4.4.39. Class Methods – N/A for Requirement 2.6.2 369](#_Toc464574961)

[6.2.4.4.40. Class Properties – N/A for Requirement 2.6.2 369](#_Toc464574962)

[6.2.4.4.41. Uses Clause – N/A for Requirement 2.6.2 369](#_Toc464574963)

[6.2.4.4.42. Forms – N/A for Requirement 2.6.2 369](#_Toc464574964)

[6.2.4.4.43. Functions 369](#_Toc464574965)

[6.2.4.4.44. Dialog – N/A for Requirement 2.6.2 397](#_Toc464574966)

[6.2.4.4.45. Help Frame – N/A for Requirement 2.6.2 397](#_Toc464574967)

[6.2.4.4.46. HL7 Application Parameter – N/A for Requirement 2.6.2 397](#_Toc464574968)

[6.2.4.4.47. HL7 Logical Link – N/A for Requirement 2.6.2 397](#_Toc464574969)

[6.2.4.4.48. COTS Interface – N/A for Requirement 2.6.2 397](#_Toc464574970)

[6.3. Network Detailed Design 397](#_Toc464574971)

[6.4. Security and Privacy 397](#_Toc464574972)

[6.4.1. Security 397](#_Toc464574973)

[6.4.1.1. Security Control Families 397](#_Toc464574974)

[6.4.2. Security Management 397](#_Toc464574975)

[6.4.3. General Security 397](#_Toc464574976)

[6.4.4. Remote Systems 398](#_Toc464574977)

[6.4.5. Contingency Planning 398](#_Toc464574978)

[6.4.6. Interfacing 398](#_Toc464574979)

[6.4.7. Electronic Signatures 398](#_Toc464574980)

[6.4.8. Security Keys 398](#_Toc464574981)

[6.4.9. Privacy 398](#_Toc464574982)

[6.5. Service Oriented Architecture / ESS Detailed Design 398](#_Toc464574983)

[6.5.1. Service Description for Virtual VA 398](#_Toc464574984)

[6.5.2. Service Design for Virtual VA 399](#_Toc464574985)

[6.5.3. Service Description for VLER/DAS 399](#_Toc464574986)

[6.5.4. Service Design for Virtual VA 399](#_Toc464574987)

[6.5.5. Service Description for VistA Web Server 399](#_Toc464574988)

[6.5.6. Service Design for VistA Web Server 399](#_Toc464574989)

[6.5.7. Service Description for HealtheVet Web Services Client 399](#_Toc464574990)

[6.5.8. Service Design for HealtheVet Web Services Client 399](#_Toc464574991)

[7. External System Interface Design 400](#_Toc464574992)

[7.1. Interface Architecture 401](#_Toc464574993)

[7.2. Interface Detailed Design 412](#_Toc464574994)

[8. Human-Machine Interface 423](#_Toc464574995)

[8.1. Interface Design Rules 423](#_Toc464574996)

[8.2. Inputs 423](#_Toc464574997)

[8.2.1. Logging onto CAPRI 424](#_Toc464574998)

[8.2.2. VistA Terminal 424](#_Toc464574999)

[8.3. Using the Software 424](#_Toc464575000)

[8.3.1. Patient Selector Screen or Patient Entry / Selection 424](#_Toc464575001)

[8.3.2. Selecting a Patient / Veteran 424](#_Toc464575002)

[8.3.3. Sensitive Records 427](#_Toc464575003)

[8.3.4. Entering a New Patient / Veteran 428](#_Toc464575004)

[8.4. Outputs 431](#_Toc464575005)

[8.4.1. C&P Exam Requests 431](#_Toc464575006)

[8.4.1.1. View/Edit Selected Request 431](#_Toc464575007)

[8.4.2. 7131 Request 433](#_Toc464575008)

[8.4.2.1. Add a New 7131 Request 434](#_Toc464575009)

[8.4.2.2. 7131 Request Status Inquiry 438](#_Toc464575010)

[8.4.3. CAPRI Reports (Patient-Specific) 439](#_Toc464575011)

[8.4.3.1. Patient Inquiry 439](#_Toc464575012)

[8.4.3.2. Detailed Inpatient Inquiry 439](#_Toc464575013)

[8.4.3.3. C&P Exam Detail 439](#_Toc464575014)

[8.4.3.4. 7131 Detal 439](#_Toc464575015)

[8.4.3.5. Additional Treating Facilities 439](#_Toc464575016)

[8.4.3.6. View Registration Data 440](#_Toc464575017)

[8.4.3.7. Patient Profile Medical Administration Service (MAS) (Full) 440](#_Toc464575018)

[8.4.3.8. Surgery Report 440](#_Toc464575019)

[8.4.3.9. Other Patient-Specific Reports 440](#_Toc464575020)

[8.4.3.10. Reprint a 21-Day Certificate 440](#_Toc464575021)

[8.4.3.11. Reprint a Notice of Discharge 440](#_Toc464575022)

[8.4.4. CAPRI Reports (Non-Patient-Specific) 440](#_Toc464575023)

[8.4.4.1. C&P Exam Reports 441](#_Toc464575024)

[8.4.4.1.1. AMIS 290 441](#_Toc464575025)

[8.4.4.1.2. AMIS 290 by Division 441](#_Toc464575026)

[8.4.4.1.3. Insufficient Exam Report 441](#_Toc464575027)

[8.4.4.1.4. Pending C&P Exams 441](#_Toc464575028)

[8.4.4.1.5. C&P Exams Checklist 442](#_Toc464575029)

[8.4.4.1.6. Re-Print C&P Final Report(s) 442](#_Toc464575030)

[8.4.4.1.7. Exam Requests by Date Range 442](#_Toc464575031)

[8.4.4.1.8. Request Status by Date Range 442](#_Toc464575032)

[8.5. Navigation Hierarchy 443](#_Toc464575033)

[8.5.1. Start – CAPRI Select a Patient 445](#_Toc464575034)

[8.5.2. Start – CAPRI Create a New Patient 445](#_Toc464575035)

[8.5.3. Create a Compensation and Pension Request 446](#_Toc464575036)

[8.5.4. Create a New 7131 Request 447](#_Toc464575037)

[8.5.5. Reports 449](#_Toc464575038)

[8.5.6. Cancellations 450](#_Toc464575039)

[9. Attachment A – Approval Signatures 456](#_Toc464575040)

[A. Additional Information 457](#_Toc464575041)

[A.1. Identification of Technology and Standards 457](#_Toc464575042)

[A.2. Constraining Policies, Directives and Procedures 459](#_Toc464575043)

[A.3. Requirements Traceability Matrix 459](#_Toc464575044)

[A.4. Packaging and Installation 459](#_Toc464575045)

[A.5. Design Metrics 459](#_Toc464575046)

# Introduction

The Compensation and Pension Record Interchange (CAPRI) project is an information

technology initiative to improve service to disabled veterans by promoting efficient

communication between the Veterans Health Administration (VHA) and Veterans Benefits

Administration (VBA). Online access to medical data enhances the timeliness of the benefits

determination. Previous attempts to automate this process were hindered by the "roll and scroll"

nature of the VHA computer interface of the Automated Medical Information Exchange (AMIE)

II. The CAPRI software acts as a bridge between the VBA and VHA information systems. It

offers VBA Rating Veteran Service Representatives and Decision Review Officers help in

building the rating decision documentation through online access to medical data. It also

offers VHA Compensation and Pension (C&P) staff an easy, standardized way of recording

C&P Examination reports.

Initially developed specifically for VBA, the utility of CAPRI has been expanded to other user

groups that include VHA, Office of the Medical Inspector, Office of Information (OI), Research,

and Veteran Service Officers. Recently, most of the newest features of CAPRI are specifically targeted at adding features to be used by VHA C&P providers and staff.

## Scope

All documents and plans related to this project are located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below.

[*http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active*](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

## User Profiles

CAPRI’s user community is primarily comprised of Veterans Benefits Administration (VBA) and Veterans Health Administration (VHA) personnel. There are also Department of Defense (DoD), VA Office of Inspector General and Veteran Service Organizations (VSO).

# Background

## Overview of the System

The Compensation and Pension Record Interchange (CAPRI) project is an information

technology initiative to improve service to disabled veterans by promoting efficient

communication between the Veterans Health Administration (VHA) and Veterans Benefits

Administration (VBA). Online access to medical data enhances the timeliness of the benefits

determination. CAPRI delivers leading edge "point and click" technology to the users' desktops.

CAPRI builds upon existing VHA information security approaches. In addition to using

established mechanisms to ensure only authorized access to medical data, CAPRI adds a level

of security by allowing VBA users to read but not alter electronic medical record information.

CAPRI also provides innovative improvements for medical centers by integrating highly detailed

(C&P) Rating examination results into the veterans' medical records.

Initially developed specifically for VBA, the utility of CAPRI has been expanded to other user

groups that include VHA, Office of the Medical Inspector, Office of Information (OI), Research,

and Veteran Service Officers. Recently, most of the newest features of CAPRI are specifically targeted at adding features to be used by VHA C&P providers and staff.

## Overview of the Business Process

All documents and plans related to this project are located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below.

[*http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active*](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

Figure 2‑1: Business Processes Diagram

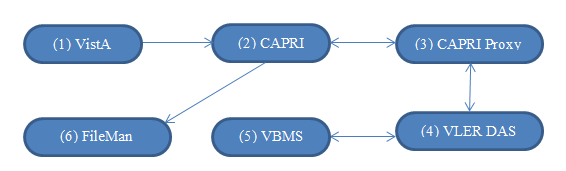


Table 2-1: Business Process

| Business Process ID | Business Process Name | Type | Owner | Description |
| --- | --- | --- | --- | --- |
| 1 | VistA | Existing | VistA | VistA is linked to CAPRI as the backbone of the system and CAPRI is the front end application |
| 2 | CAPRI | Existing | VistA/Fileman | The first functionality is that a C&P exam is created in CAPRI for a patient. Second, the DBQ associated to the exam is generated and completed. |
| 3 | CAPRI Proxy | Existing | CAPRI | Web service used by the CAPRI to forward a DBQ to VLER DAS using two-way SSL. |
| 4 | VLER DAS | Existing | CAPRI | The logic of CAPRI that transmits files to VLER DAS |
| 5 | VBMS | Existing | VBMS | XML and PDF file of exam results is sent to VBMS where it can be read as an item associated to the patient. |
| 6 | Fileman | Existing | VistA | XML is now transmitted to the 2507 Exam file (396.4) and the exam stores the data in XML format. This data is associated to the exam. |

## Overview of the Significant Requirements

All documents and plans related to this project are located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below.

[http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

Table 2-2: Functional Requirements

| ID | Requirement |
| --- | --- |
| ID # 608696 | After BHIE has been disabled, the DoD tab in CAPRI will be disabled and access to a Service Member or Veteran’s DoD information will occur within a VistA Web tab so that the information is available for defining or updating exam requests or requests for medical services. |
| ID # 513608 | As a userinvolved with the disability claims/exam request process (C&P Exam Requests) improvements are required to the scheduling, routing and processing of exam requests so that Veteran claims and requests can be processed without delays in care or fulfillment. |

### Overview of Significant Functional Requirements

Please refer to the BRD referenced above. There aren’t any significant impacts on design for this release. The changes listed above augment existing functionality, with many changes making in place modifications such as adding an extra option to a selection dialog, modifying the contents presented to a user.

### Overview of Functional Workload / Performance Requirements

The specific enhancements for this release will have no impact to functional workload / performance requirements and no proxy server code has been changed.

As it pertains to the production servers, on a typical day we see an average number of transactions per hour range at about 105 with a maximum of about 960.  This includes posts of exam requests and queries of exam results.  Transactions, on average, take a few seconds to complete.  During a 24 hour period, we may see an error count of around 25 including proxy authentication failed or proxy authentication timeout. Load testing was conducted at 900 transactions per minute.

Table 2-3: Load Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Average # Transactions per hour | Max # Transactions per hour | Average Response Time (ms) | Max Response Time (ms) | Average errors per 24 hour period | Memory Usage | CPU Usage Average | CPU Peak |
| 105 | 960 | 2268 | 50222 | 21.667 | < 1 GB | < 20% | 65% |

### Overview of Operational Requirements

| ID | Requirement |
| --- | --- |
| REQ. 2.6.1  **Business Requirement 513608** | As a user involved with processing exam requests (C&P Exam Requests) I need to be able to route/re-route requests at the intra-facility, inter-facility, regional and enterprise-wide levels, so that the request does not have to be cancelled and re-initiated. |
| REQ. 2.6.2  **Business Requirement 608794** | As a user involved with managing the processing of exam requests, I need to manually activate new C&P routing locations identified within VistA, so that users processing requests cannot select invalid routing locations that do not process the C&P Exam requests. |
| REQ. 2.6.3  **Business Requirements 608696** | As a user processing requests (e.g., after Bidirectional Health Information Exchange (BHIE) has been disabled), I need to access a Service Member or Veteran’s DoD information (e.g., within a VistA Web tab) so that the information is available for defining or updating exam requests or requests for medical services. |

### Overview of the Technical Requirements

The technical requirements for this release are not different from those of the existing system.

### Overview of the Security or Privacy Requirements

CAPRI has an active Privacy Threshold Analysis (PTA) on file.

### Overview of System Criticality and High Availability Requirements

This modification does not make any modifications to the criticality and availability requirements for Legacy VistA System. CAPRI falls under VistA disaster recovery.

### Single Sign-on Requirement

There is no modification to the single sign-on requirements. CAPRI uses Legacy VistA authentication.

### Requirement for Use of Enterprise Portals

CAPRI does not use Enterprise Portals

### Special Device Requirements

This release shall comply shall comply with all Federal Accessibility regulations—Currently falling under Section 508 of the U.S. Rehabilitation Act (29 U.S.C. § 794d). Full details on requirements for compliance with Section 508 can be found at [www.section508.gov](http://www.section508.gov/).

# Conceptual Design

## Conceptual Application Design

### Application Context

There are no application context changes for this release. CAPRI application interacts with other systems as described in the diagram Figure 3‑1.CAPRI utilizes multiple interfaces from external applications/systems. These include VistA RPC, WebServices, Secure FTP, Shell execute calls for Windows and Interprocess communication through CCOW RPC Broker. Currently Virtual VA and VLER/DAS are connected using Webservices to CAPRI GUI client application. CAPRI VistA application connects to MVI using HealtheVet Webservice VistA client on the VistA server side. Single context sign-on communication is provided between CAPRI and CPRS as well as any other Windows Client application utilizing RPC Broker on the client workstation. Single context sign-on messaging is also utilized between one authenticating VistA/CAPRI server and other VistA/CAPRI servers within VA’s network via a triangulating multi-part token messaging mechanism using RPC Broker’s Broker Security Enhancement features. CAPRI also utilizes third party applications Adobe Acrobat and Microsoft Office Document Imaging as external document viewers and uses OLE automation to control Microsoft Word as a document editor.

Figure 3-1: Application Context Diagram



Table 3-1: Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| 1 | VistA database connection | Primary communication channel between CAPRI GUI and VistA server | RPC Broker | CAPRI GUI and Server interconnect |
| 2 | Third party Exam Vendors | Contracted Vendors processing Exam Requests sent by Providers and providing results for those requests. | SFTP | CAPRI |
| 3 | Virtual VA | Storage and retrieval of patient electronic documents | Webservice | CAPRI |
| 4 | VLER/DAS | Storage and retrieval of exam results for both internal to the VA exams and external vendors. | Webservice | CAPRI |
| 5 | External Document Editor | Third party application MS-Word | OLE Automation | CAPRI |
| 6 | External Document Viewers | Third party viewers for PDF and TIFF files using Adobe Acrobat and MS Office Document Imaging respectively. | Shellexecute | CAPRI |
| 7 | CCOW RPC Broker | Single sign-on capability to allow users to connect with VistA servers on one CCOW enabled application and then users can omit having to supply credentials again for any other CCOW enabled application | Interprocess communication | CAPRI, CPRS and others |
| 8 | HealtheVet | Perform MVI Search from a VistA server using HealtheVet webservice. | Webservice | CAPRI |

Table 3-2: Interfaces External to OIT

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| 1 | RPC Broker | Primary communication channel between CAPRI GUI and backend VistA server data storage. | CAPRI GUI accepts user input for multiple business functions it supports and translates it into appropriate VistA FileMan commands and data streams to transmit those to a backend VistA server.  CAPRI also processes VistA server responses and appropriately presents those to the user via the GUI interface. | VistA server interprets all communication from CAPRI GUI and appropriately handles the requests.  Requests include but aren’t limited to User authentication, patient related information search, add, update and delete (CRUD). | N/A |
| 1a | RPC Broker | Special communication channel specifically for multi-server authentication requests | CAPRI GUI requests security tokens from an authenticating server and presents those to a new server for which “authentication” is required. Handles the resulting response from the “authentication” server. | VistA server processes security tokens and authenticating server information and makes a backend connection to validate the token and authentication information provided. Provides a response to the GUI. | N/A |
| 3 | Webservice | Virtual VA | CAPRI sends patient electronic reports for Storage in Virtual VA. CAPRI also may request a list of documents available per patient and request retrieval of patient’s documents. | Virtual VA provides confirmation of document storage.  VVA also provides documents in response to retrieval requests. | Virtual VA |
| 4 | Webservice | VLER/DAS | CAPRI sends patient exam results using XML to VLER/DAS. CAPRI may also request a list of exam results. A single exam may be viewed and reviewed for signature. | VLER/DAS provides confirmation of document storage. VLER/DAS provides documents in response to retrieval requests. | VLER/DAS -> VBMS |
| 5 | External Editor | MS-Word | CAPRI creates a draft Word Document file with pre-filled information and launches MS-word to allow user to edit and save this document. The document is then sent to the vendors. |  | N/A |
| 6 | Document Viewers | Adobe Acrobat  Microsoft Office Document Imaging | Third party apps are launched as viewers for documents (retrieved from Virtual VA) | N/A | N/A |
| 7 | CCOW RPC Broker | CPRS and other applications | Single sign-on functionality | Authentication information is exchanged between applications through interprocess communication | N/A |
| 8 | HealtheVet webservice | MVI Search webservice | Patient query criteria | Zero or more patient query results | MVI |

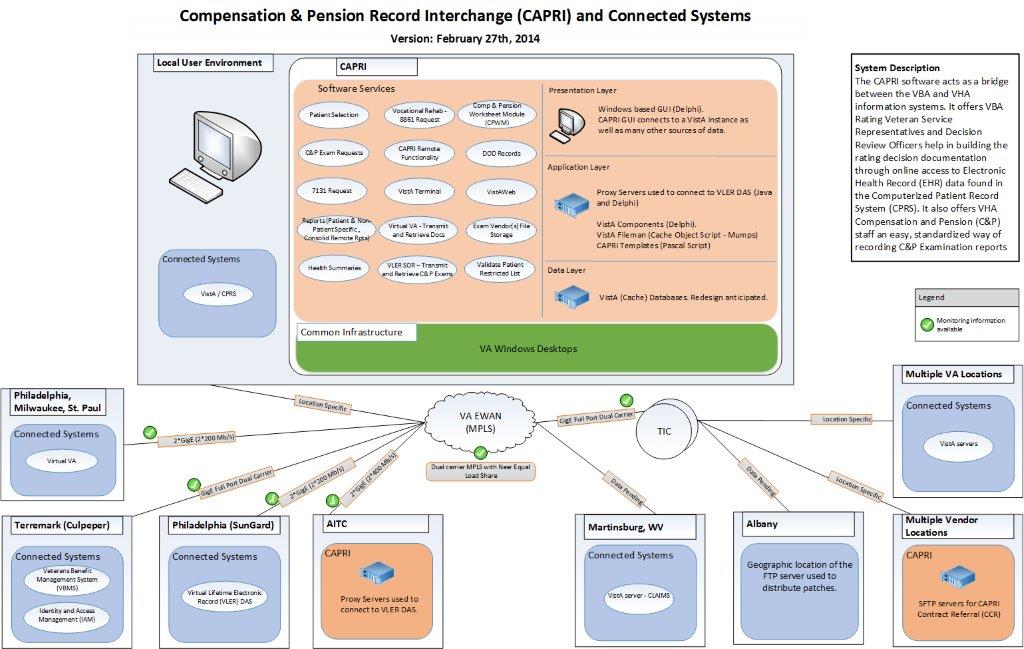
Table 3-3: Externally Shared Data Stores

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Data Stored | Owner | Access |
| N/A | N/A | N/A | N/A | N/A |

### High-Level Application Design

There are no application design changes for this release. Enhancements will be accomplished within the design framework of the current CAPRI application.

Figure 3-4: CAPRI High-Level Application Design



### Application Locations

There are no changes to the Application Locations for this release. CAPRI is a desktop application used at all VA Medical Centers. The application acts as a GUI front-end for the VistA AMIE application. The CAPRI application interfaces with the local VA Medical Center VistA system database.

Table 3-5: Application Users

| Application Component | Location | User |
| --- | --- | --- |
| C & P Exam Requests | VA sites nationwide | VBA C & P Examiners |
| DBQ Processing | VHA facilities | VHA Personnel |
| Reports | VBA Regional Offices and VA Medical Centers | VBA and C&P Staff |
| VistA button | Remote Users (Regional Offices or VA Medical Centers) | HRC Staff |
| Vocational Rehab Tab | VBA Regional Offices and VA Medical Centers | VBA and DHN So ational Rehab staff |
| CAPRI Contract Referral | VA Medical Centers | C&P Staff |
| CAPRI Remote Menus | Nationwide VBA, HIA | VBA and HIA Staff |

## Conceptual Data Design

### Project Conceptual Data Model

There are no changes to Conceptual Design for this release. The CAPRI application utilizes the underlying data model of the VistA system. For details of the VistA system please refer to relevant documents on VDL website, under “Financial and Administrative”, “Automated Medical Information Exchange (AMIE)” package. The direct link is listed below.

[http://www.DNS /vdl/application.asp?appid=31](http://www.DNS   /vdl/application.asp?appid=31)

### Database Information

The CAPRI application utilizes the underlying data model of the VistA system. For details of the VistA system please refer to relevant documents on VDL website, under “Financial and Administrative”, “Automated Medical Information Exchange (AMIE)” package. The direct link is listed below.

[http://www.DNS /vdl/application.asp?appid=31](http://www.DNS   /vdl/application.asp?appid=31)

### User Interface Data Mapping

There will be no significant changes to the Interface Data Mapping model currently used by the CAPRI application. New fields may be added to existing VistA files in the AMIE name space as required to support the enhancements listed in the RSD for the Patch. These additions will depend on the number of business priorities implemented with available funding. The User Manual will be updated to reflect any new screens, or data elements on input screens and reports.

#### Application Screen Interface

There will be minor changes to existing screens of the application GUI to accommodate new data fields listed in the RSD for this Patch. The specific changes made will depend on business priorities. The User Manual will be updated to reflect any new screens.

#### Application Report Interface

There will be no significant changes to existing reports provided by the application to accommodate new data fields that may be listed in the RSD. The specific changes made will depend on business priorities the User Manual will be updated to reflect any new data elements on reports.

#### Unmapped Data Element

No unmapped data elements apply to this project.

## Conceptual Infrastructure Design

The CAPRI application utilizes the underlying Infrastructure of the VistA system.

### System Criticality and High Availability

The CAPRI system is a GUI front-end to VA Medical Center VistA systems. CAPRI complies with the VistA system availability standards of 99.9% availability, outside of regularly scheduled down time for maintenance.

### Special Technology

There is no special technology for this project.

### Technology Locations

The CAPRI application utilizes the underlying Infrastructure of the VistA system.

### Conceptual Infrastructure Diagram

For the Conceptual Infrastructure Diagram, refer to Figure 3-1.

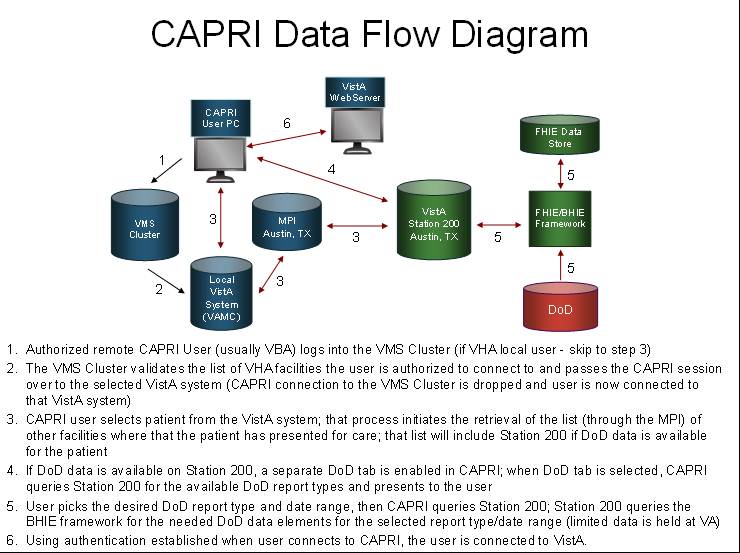
# System Architecture

The CAPRI application utilizes the underlying Infrastructure of the VistA system. The CAPRI system is a GUI front-end to VA Medical Center VistA systems. As such, it will comply to the VistA system availability standards of 99.9% availability, outside of regularly scheduled down time for maintenance.

## Hardware Architecture

The diagram illustrates a high-level association of CAPRI and the systems it interfaces with.

Figure 4-1: CAPRI Data Flow Diagram



CAPRI’s hardware infrastructure is distributed per the table below. It consists of a VistA Station 200, MPI, CAPRI User PC (Thousands of VA Workstations nationwide), Two (2) CAPRI Authentication/Proxy Server and depends on the Local VistA System (VAMC), VistA Web Server, FHIE Data Store, and the FHIE/BHIE Framework as described in the following table:

Table 4-1: Hardware Infrastructure

| Hardware | Qty | Centralized (y/n) | Location Support (OIFO) | Type (workstation/server) | OS (Operating System) |
| --- | --- | --- | --- | --- | --- |
| VistA Station 200 |  |  | Austin, TX |  |  |
| MPI |  |  | Austin. TX |  |  |
| CAPRI User PCs (VA Workstations) | 10,000+ | No | Nationwide | Workstation | Windows 7 |
| CAPRI Authentication/ Proxy Server | 2 | Yes | Austin, TX | Cluster of Servers | Windows Server 2008 |
| Additional Hardware Dependencies from VistA | | | | | |
| Local VistA System (VAMC) |  |  | Nationwide |  |  |
| VistA Web Server |  |  | Nationwide |  |  |
| FHIE Data Store |  |  | Nationwide |  |  |
| FHIE/BHIE Framework |  |  | Nationwide |  |  |

System: OS OpenVMS V8.3 for VistA, CAPRI GUI does not utilize this Operating System.

NCRA01, hp AlphaServer ES80 7/1150

NCRA02, hp AlphaServer ES80 7/1150

NCRA03, hp AlphaServer ES47 7/1150

NCRA04, hp AlphaServer ES47 7/1150

HP EVA8000 SANS Storage Array

HP MSL6000 Tape Library

## Software Architecture

There are no software architecture changes for this release. CAPRI’s software is comprised of the following programs:

System: OS OpenVMS V8.3 for VistA, CAPRI GUI does not utilize this Operating System.

Cache for OpenVMS/ALPHA V8.x (Alpha) 2011.1.2 (Build 701 + Adhoc 12942) 19-JUL-2013

Table 4-2: Software Infrastructure

| **Software Inventory** | **Qty** | **Location Support (OIFO)** | **Hosting Server (Hardware)** | **Operating System** | **Database** | **Compilers** | **Communi-cation Software** | **Tools** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **VistA/ Fileman** | 135+ | Nationwide | Various | Linus and VMS | CACHE |  |  |  |
| **CAPRI GUI Suite** | 30,000+ | Nationwide | Various | Windows 7 |  | Delphi |  | Rational Team Concert |
| **CAPRI Proxy** | 2 | Austin, TX | Virtual Servers | Windows 2008 Server |  | Java | WebLogic | Rational Team Concert |
| **CAPRI Authentication Module** | 2 | Austin, TX | Virtual Servers | Windows 2008 Server |  | Delphi |  | Rational Team Concert |

## Network Architecture

There are no network architecture changes for this release. CAPRI GUI communicates with various servers within the VA intranet via TCP/IP. Existing VA intranet WAN backbone is leveraged as needed by any software component or service within the application. Type of traffic over TCP/IP includes, RPCBroker, HTTP and HTTPS. Network architecture is displayed in Figure 3-2.

## Service Oriented Architecture / ESS

There have been no changes to the service oriented architecture in this release.

### Services Provided

#### DAS

The DAS Health program gives VA and non-VA health care providers secure access to certain parts of a veteran’s electronic health record. When VA health care exams are completed, clinical exam results are sent to DAS from CAPRI to be distributed to other systems. In addition, CAPRI sends requests for clinical exams to non-VA health care providers via DAS.

#### Virtual VA

### CAPRI sends patient electronic reports for storage in Virtual VA.

### Services Required / Consumed

#### DAS

CAPRI may request a list of exam results from non-VA health care providers. A single exam may be viewed and reviewed for signature.

#### Virtual VA

CAPRI may request a list of documents available per patient and request retrieval of patient’s documents.

#### HealthEVet Web Service

CAPRI consumes services from HealthEVet to provide access to the MVI for an enterprise veteran search.

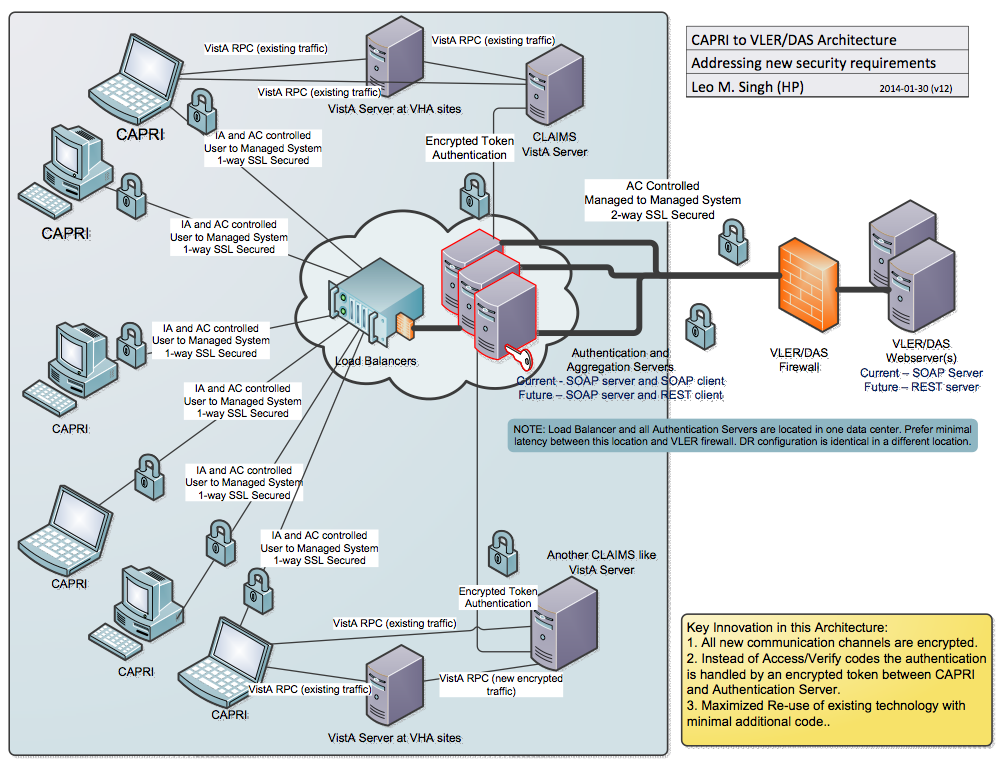
## Enterprise Architecture

The current TRM/SP is located VA Enterprise Architecture (EA) v2.1 at [http://DNS .DNS /](http://DNS    .DNS   /).

The OneVA EA HOME page is located at[*http://DNS ea.oit.DNS /*](http://DNS  ea.oit.DNS   /)

The Capri Enterprise Archectecture complies with the OneVA EA Standard.Click [here](http://DNS  DNS       .DNS   /development/legacy_product_enhancements/healthcare_management_products/CAPRI/Project%20Manager%20Resources/Enterprise%20Compliance%20Criteria/OneVA%20EA%20_ETA%20Compliance_CAPRI%20Tactical%20Enhancements_12032012.docx) for access to the form.

Figure 4-3: CAPRI to DAS Architecture



**Table 4-3 CAPRI Methodology and Tools List**

| Discipline | Technology Name | Owner/Host | Version | TRM Decision | Waiver |
| --- | --- | --- | --- | --- | --- |
| Project Management | Primavera P6 | VA Supplied Tool Scheduling Team | 7.0.5.5 | Approved | N/A |
| Requirements Management | Rational DOORS Next Generation (RDNG) | VA Supplied Tool, Rational Tools Team | 6.0 | Approved | N/A |
| Configuration Management | Rational Team Concert (RTC) | VA Supplied Tool, Rational Tools Team | 6.0 | Approved | N/A |
| Configuration Management | Rational Team Concert (RTC) Eclipse Client | VA Supplied Tool, Rational Tools Team | 6.0.1 | Approved | N/A |
| Quality Management | Rational Quality Manager (RQM) | VA Supplied Tool, Rational Tools Team | 6.0 | Approved | N/A |
| Development | Delphi | CAPRI | 2006 / XE2 | Not Approved | Delphi upgrade is in process and scheduled for deployment by June 2017. |
| Infrastructure | Windows Enterprise and VMware | AITC Hosting Center | 2008 R2 | Approved | N/A |
| Infrastructure | WebLogic 12c | AITC Hosting Center | 12c | Approved | N/A |
| Infrastructure | Open SSL | AITC Hosting Center | 1.0.1i | Approved | N/A |
| CAPRI Development | XML | CAPRI | 1.0 (5th edition) | Approved | N/A |
| Infrastructure | SFTP | AITC Hosting Center | 10 | Approved | N/A |
| Development | MUMPS |  | 2014.1.3 | Approved | N/A |
| Infrastructure | Adobe Acrobat  Microsoft Office Document Imaging  Microsoft Word | VA Desktop Applications | N/A | N/A  CAPRI utilizes the version of the document as it is received. CAPRI does not convert or update the version of the document. | N/A |
| Infrastructure | Cache for OpenVMS/ALPHA V8.x (Alpha) 2014.1.3 | VistA Hosted Infrastructure | N/A | Approved | N/A |

# Data Design

This section outlines the design of the database management system (DBMS) and non-DBMS files associated with the system.

## DBMS Files

There will be minor changes to the VistA database to accommodate new data fields listed in the RSD for this release. All database schema updates are approved by the VistA DBA before implementation.

## Non-DBMS Files

There are no Non-DBMS files being used in the CAPRI application.

## Data View

Description of the VistA DataView can be found in theVistA Monograph at [http://www.ehealth.DNS /docs/VistA\_Monograph.docx](http://www.ehealth.DNS   /docs/VistA_Monograph.docx)

The following excerpt from this document is an introduction to VistA Monograph: “*This Monograph* provides an overview of the **V**ETERANS HEALTH **I**NFORMATION **S**YSTEMS and **T**ECHNOLOGY **A**RCHITECTURE information system – **VistA** – used by the Veterans Health Administration (VHA) of the U.S. Department of Veterans Affairs (VA) in serving America’s veterans through the provision of quality health care which enhances our Veterans’ health and well-being.

# Detailed Design

## Hardware Detailed Design

CAPRI (VistA/AMIE) utilizes existing hardware infrastructure, VistA servers and local user Windows workstations. Hence there is no separate Hardware Detailed Design document.

CAPRI Proxy Servers are configured as follows.

* Network load balanced, F-5 (shared in the AITC data center)
* VMWare virtual machines – Qty 2
* 8 GB RAM
* 40 GB shared storage
* Windows 2008 Server R2 Enterprise Edition
* Oracle Weblogic 12c

## Software Detailed Design

This section focuses on changes and enhancements along with new features being introduced in CAPRI in the current patch.

CAPRI is a large application developed over many years. There are a large number of RSD’s that have been created over the years. This SDD document only lists the changes introduced in this release and references the relevant change control numbers. For a full list of CAPRI System Features reference CAPRI Users Guide and other CAPRI documents as well as previous versions of CAPRI SDD (System Design Documents).

### Conceptual Design

Instructions for this section call for “how the software will be built”. However CAPRI System is already in operation and in the current patch there were no “Conceptual Design” changes. Hence there is no design specific information available to provide for this section.

### Specific Requirements

#### Requirement 2.6.1

As a user involved with processing exam requests (C&P Exam Requests) I need to be able to route/re-route requests at the intra-facility, inter-facility, regional and enterprise-wide levels, so that the request does not have to be cancelled and re-initiated.

#### Database Repository

N/A. CAPRI is a front end for VistA.

#### System Features

* As a CAPRI user, I want an option titled “Re-Route Request” displayed on the C&P Exams screen that will enable the system to re-route the exam request at the request level to another facility/site.
* As a CAPRI user, I want to ensure that if an exam request has already been re-routed to another facility/site then the option to re-route will be disabled. An existing re-routed request cannot be re-routed.
* As a CAPRI user, I want the re-routing option to display the below items:
  + Destination Facility/Site (Required)
  + Re-routing Reason (Required)
  + Free text field of 250 characters for further explanation to the recipient as to the reason for the re-routing (**Optional unless a re-routing “reason” of OTHER is selected then it will be required.**)
* As a CAPRI user, I want an option titled “Accept Re-Routed Request ” displayed on the C&P Exams screen that will enable the system to accept the exam request at the request level from another facility/site.
* As a CAPRI user I want the system to process the accepted re-routed request (e.g., update status).
* As a CAPRI user, I want to ensure that an option to accept the exam request is **ONLY** available if the exam request was re-routed to my facility.
* As a CAPRI user, I want an option titled “Reject Re-Routed Request ” displayed on the C&P Exams screen that will enable the system to reject the exam request at the request level to another facility/site.
* As a CAPRI user, I want to ensure that an option to reject the exam request is **ONLY** available if the exam request was re-routed to my facility.
* As a CAPRI user, I want the system to display a free text field of 250 characters when selecting the reject option.
* As a CAPRI user, I want to enter text in the free text field so that I can explain why the re-route request is being rejected.
* As a CAPRI user I want the system to enable the submit reject option once all text has been entered in the free text field so that I can submit the rejection.
* As A CAPRI user want the free text field to be mandatory when the reject option is selected.
* As a CAPRI user I want the system to process the rejection (e.g., update status).
* As a CAPRI user I want the system to notify **the facility that re-routed the request** of the rejection information.
* As a CAPRI user I want the notification to include the following:
  + Identifier (e.g., DFN) that can be utilized for accessing the patient record on the original routing VistA instance.
  + Free text description of the reject reason. (Will be excluded from the email but will be visible in the CAPRI GUI)
* The name of the email group will be **DVBA C 2507 REROUTE.**
  + The facility will be responsible for adding members to this group who should receive information concerning the 2507 Request Reroute status.
* As a CAPRI user, for the **Destination Facility/Site** option, I want the system to display a list of facilities (names)/site (names) and location numbers to which the request can be re-routed. This will require a two-step process listed below:
  + Step 1: A list box for the facility name will display and the user will select the VistA instance.
* Once the VistA instance displays, the routing location details can be selected, retrieved and displayed in the Routing Location list box. As a CAPRI user, I want the display option to list the facilities (names)/site ( names) and location numbers in the following order:
  + Alphabetically
  + State
  + Location number
* As a CAPRI user, I do not want the facility or site options to be limited by the existing VistA instance that I may be connected to.
* As a CAPRI user I want the option to select facilities/sites within other regions and VistA instances.
* As a CAPRI user, I want the system to display re-routing reason options that can be selected as listed below:
  + INCORRECT JURISDICTION
  + VETERAN IS AN EMPLOYEE
  + VETERAN REQUEST/PREFERENCE
  + OTHER
* As a CAPRI user, I want the system to allow the re-routing reason options (Veteran is an employee, incorrect jurisdiction, Veteran request/preference, and other) as editable at the National level by central office personnel which will include the following details:
  + A new option will be added that will allow users the ability to add/inactivate/activate the list of reasons within the exam cancellation window.
  + A security key added for the new option display and shall be named “**DVBA CAPRI REROUTEREASONS**”
  + The new option will be available on the CAPRI Help menu and visible to users who have been assigned the security key.
  + The new option will display in the GUI named “Manage Routing Reasons”.
  + A new GUI screen created that will allow the user to add/inactivate/activate re-routing reasons.
* As a CAPRI user, after all re-routing options have been selected; I want the system to allow the submission of the re-routing request.
* As a CAPRI user, I want the system to re-route the request.
* As a CAPRI user, I want the system to notify the **original sender** (VBA user/forwarding email address) via email of the re-routing information. The email notification will include the following details:
  + Identifier (e.g., Data File Number [DFN]) that can be used for accessing the patient record on the original routing VistA instance.
  + Re-route destination: VistA instance (facility/site)
  + Drop down reason for re-route.
  + Free text description of re-route. (Will be excluded from the email but will be visible in the CAPRI GUI)
* As a CAPRI user, I want the patient information in the original routed record to be utilized to create a record with the re-routed VistA instance when a patient record does not exist.
* As a CAPRI user, I want this process to be automatic or in the case of data integrity issues, the process should be facility so that the user does not need to retype or copy and paste information.
* The routing information will include the following:
  + Exam request status
  + Routing information
  + Identifiers that can be used for accessing the patient record information.
* The request status will be maintained. The exam request statuses will include existing exam request statuses and new statuses to represent the following:
  + If an exam has been re-routed, the below statuses are possible on the instance **from where (Sending Facility)** the request was re-routed.
    - Re-Routed – Pending Acceptance
    - Re-Routed – Accepted \*\*When the Destination Facility accepts the request the status updates to this\*\*
    - New- (After a nighlty job runs, this status is updated to PENDING REPORTED and will include rejection comments.)
  + If an exam has been re-routed, the below statuses are possible on the instance **to which (Destination Facility)** the request was re-routed.
    - New - Re-Routed
    - Re-Routed – Pending Acceptance (After a nightly job runs, this is the updated status)
    - PENDING REPORTED (If the site Accepts the Re-Routed request)
    - Re-Routed – Rejected
* The re-routing information will be maintained and is thought to include the following:
  + - Facility/Site information indicating to where the request has been re-routed (if applicable)
    - Facility/Site information indicating from where the request was re-routed (if applicable)
    - Drop down reason for re-route (if applicable).
    - Free text description of re-route or rejection (if applicable).
    - Date information:
* If an exam has been re-routed, the following dates will be stored as applicable on the instance **from which (Sending Facility)** the request was re-routed.
  + - * + Date Re-Routed Out
        + Date Re-Route Accepted
        + Date Re-Route Rejected
* If an exam has been re-routed, the following dates will be stored as applicable on the instance **to which** **(Destination Facility)** the request was re-routed.
  + - * + Date Re-Routed In (Create Date/Time)
        + Date Re-Route Accepted
        + Date Re-Route Rejected
* As a CAPRI user, I want the system to maintain Identifiers for accessing patient record information on the original VistA instance **from which** the request was re-routed if applicable.
* As a CAPRI user, I want the system to maintain Identifiers for accessing patient record information on the VistA instance **to which** the request was re-routed (if applicable)
* As a CAPRI user, I want CAPRI and AMIE reporting that displays the status of the request to allow for the additional statuses (e.g., Re-Routed – Pending Acceptance, Re-Routed – Accepted, Re-Routed – Rejected and New – Re-routed).

**CAPRI** reports and filters that will display the additional statuses:

* + **Pending C&P Exams** report – has a status report filter on the GUI. The statuses of  Re-Routed – Accepted, and Re-Routed – Rejected will not display in the filter.
  + **Request Status by Date Range** report – has a status report filter on the GUI.
  + **C&P Exams Tab>Status Inquiry** screen, the Reports tab displays the details of the **C&P Exam Detail** report. The field named **Status of request** will need to display all new statuses when applicable.
  + **Pending C&P Exams** – Report detail in both txt and csv will need to include displaying all new statuses when applicable. The statuses of  Re-Routed – Accepted, and Re-Routed – Rejected will not display on the report.
  + **Exam Request by Date Range** - Report detail in both txt and csv will need to include displaying all new statuses when applicable.
  + **Request Status by Date Range** - Report detail in both txt and csv will need to include displaying all new statuses when applicable.

**AMIE** reports and functions that will display the additional statuses:

* **AMIE** Function
* **Edit C&P Request Information** – Function
* **AMIE** C&P Reports Menu
* **Pending C&P Exams Report**
* **AMIE** Regional OfficeReports Menu
* **Pending C&P Exams Report**

* As a CAPRI user, In the case of re-routed statuses, I want the report to indicate the re-routed location and the date on which it was re-routed.
* As a CAPRI user, for all existing CAPRI and AMIE reporting for which the status can be utilized as a report filter, I want the additionally defined statuses to be listed within the statuses that can be selected.
* As a CAPRI user, for all existing CAPRI and AMIE reporting for which a resolved column exists and for the status inquiry screen, I want the resolved column to indicate that the request was resolved on the VistA instance if the status of the request is **re-routed - accepted**.
  + - * **Requirement:** New fields shall be added in the Standard Data Dictionary #396.34 for the 2507 Re-Route Exam Request process to support informatics reporting in the CDW (e.g., tracking re-route rejection rates for an office that has re-routed to more than one VistA). **(Business Requirements 606458)**
* Requirement: As a CAPRI user, I want the system to provide the ability to locally re-route a 2507 Exam Request so that I can re-route within the same VAMC/VistA Instance.

#### Design Element Tables

##### Routines (Entry Points)

Table 15 (Grouping): Routines

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBAB1C | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1, 2.6.1.2 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBCUTL8  ^DVBCXFR2 | ^DIC  $$EXTERNAL^DILFD  $$SITE^VASITE  $$FMTE^XLFDT  ^XMD  $$GOTLOCAL^XMXAPIG |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DVB(396.4  ^DVB(396.6  ^TMP(“XMERR”  ^TMP($J  ^VA(200  ^XMB(3.8 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **DVBAB1C ;AJF - CAPRI UTILITIES ; 09/13/2016 3:52pm**  **;;2.7;AMIE;\*\*193\*\*;Apr 10, 1995;Build 15**  **;**  **Q**  **;**  **MSG(RIEN) ;Generate mail message;AJF**  **;**  **D GETREQ**  **D MGUSR**  **;**  **S XMSUB="CAPRI: 2507 Exam Request Rejected"**  **S ^TMP($J,"AMIE1",1)="The 2507 Exam Request as described below has been**  **rejected."**  **S ^TMP($J,"AMIE1",2)=""**  **S ^TMP($J,"AMIE1",3)=""**  **S ^TMP($J,"AMIE1",4)=" DFN: `"\_DVBADFN**  **S ^TMP($J,"AMIE1",5)=" Requested Date: "\_DVBADT**  **S ^TMP($J,"AMIE1",6)=" Requested Site: "\_RTF**  **S ^TMP($J,"AMIE1",7)=""**  **S ^TMP($J,"AMIE1",8)=" Rerouted Date: "\_RDT**  **S ^TMP($J,"AMIE1",9)=" Rerouted Site: "\_RTO**  **S ^TMP($J,"AMIE1",10)=""**  **S ^TMP($J,"AMIE1",11)=""**  **S ^TMP($J,"AMIE1",12)=""**  **S ^TMP($J,"AMIE1",13)="\*\*NOTE: To view the patient using the DFN, paste**  **the DFN number into the CAPRI"**  **S ^TMP($J,"AMIE1",14)="Patient Selector 'Patient ID' field to find the**  **patient. Be sure to include"**  **S ^TMP($J,"AMIE1",15)="the ` (backward-apostrophe) character."**  **S ^TMP($J,"AMIE1",16)=""**  **S ^TMP($J,"AMIE1",17)=""**  **S ^TMP($J,"AMIE1",18)=""**  **S ^TMP($J,"AMIE1",19)="\*\*\*\*\*This is an auto-generated email. Do not re**  **spond to this email address.\*\*\*\*\*"**  **S XMTEXT="^TMP($J,""AMIE1"","**  **D ^XMD,END**  **Q**  **;**  **;**  **FINDEXAM(ZMSG,ZIEN) ;Returns list of exams in 396.4 that are linked to ZIEN in 3**  **96.3**  **N DVBABCNT,DVBABIEN**  **S DVBABCNT=0,DVBABIEN=0**  **F S DVBABIEN=$O(^DVB(396.4,"C",ZIEN,DVBABIEN)) Q:'DVBABIEN D**  **.S DVBABD1=$P($G(^DVB(396.4,DVBABIEN,0)),"^",2)**  **.S DVBABD2=$P($G(^DVB(396.6,+$P($G(^DVB(396.4,DVBABIEN,0)),"^",3),0)),"**  **^",1) ;Name of Exam**  **.S DVBABD3=$P($G(^DVB(396.4,DVBABIEN,0)),"^",4)**  **.I DVBABD3="O" S DVBABD3="[OPEN]"**  **.I DVBABD3="C" S DVBABD3="[COMPLETE]"**  **.I DVBABD3="X" S DVBABD3="[CANCELED BY MAS]"**  **.I DVBABD3="RX" S DVBABD3="[CANCELED BY RO]"**  **.I DVBABD3="T" S DVBABD3="[TRANSFERRED OUT]"**  **.I ZIEN=DVBABD1 D**  **..S ZMSG(DVBABCNT)=DVBABIEN\_"^"\_DVBABD2\_" "\_DVBABD3**  **..S DVBABCNT=DVBABCNT+1**  **K DVBABCNT,DVBABIEN,ZIEN,DVBABD1,DVBABD2,DVBABD3**  **Q**  **;**  **;**  **SENDMSG(RIEN) ;SET UP TO SEND EMAIL/NOTIFICATION TO REQUESTOR OF 2507**  **D GETREQ**  **Q:DVBAEA=""**  **;**  **RDYMSG ;SEND REROUTED MESSAGE TO REQUESTOR OF 2507**  **;no text/body is passed in so we have to build the message from scratch**  **S XMSUB="CAPRI: 2507 Exam Request Rerouted"**  **S ^TMP($J,"AMIE1",1)="The 2507 Exam Request as described below has been**  **rerouted."**  **S ^TMP($J,"AMIE1",2)=""**  **S ^TMP($J,"AMIE1",3)=""**  **S ^TMP($J,"AMIE1",4)=" DFN: `"\_DVBADFN**  **S ^TMP($J,"AMIE1",5)=" Requested Date: "\_DVBADT**  **S ^TMP($J,"AMIE1",6)=" Requested Site: "\_RTF**  **S ^TMP($J,"AMIE1",7)=""**  **S ^TMP($J,"AMIE1",8)=" Rerouted Date: "\_RDT**  **S ^TMP($J,"AMIE1",9)=" Rerouted Site: "\_RTO**  **S ^TMP($J,"AMIE1",10)=""**  **S ^TMP($J,"AMIE1",11)=" Reroute Reason: "\_RRR**  **S ^TMP($J,"AMIE1",12)=""**  **S ^TMP($J,"AMIE1",13)=""**  **S ^TMP($J,"AMIE1",14)="\*\*NOTE: To view the patient using the DFN, paste**  **the DFN number into the CAPRI"**  **S ^TMP($J,"AMIE1",15)="Patient Selector 'Patient ID' field to find the**  **patient. Be sure to include"**  **S ^TMP($J,"AMIE1",16)="the ` (backward-apostrophe) character."**  **S ^TMP($J,"AMIE1",17)=""**  **S ^TMP($J,"AMIE1",18)=""**  **S ^TMP($J,"AMIE1",19)=""**  **S ^TMP($J,"AMIE1",20)="\*\*\*\*\*This is an auto-generated email. Do not re**  **spond to this email address.\*\*\*\*\*"**  **S XMTEXT="^TMP($J,""AMIE1"","**  **D ^XMD,END**  **Q**  **;**  **GETREQ ; Get infor the RIEN**  **N DVBA0,DVBAREQ,DVBAC,DVBAQUIT,DUZ**  **N MSG,MERR,CTR**  **;SINCE MAILMAN DOES NOT ALLOW MESSAGES TO BE SENT FROM USERS WITHOUT AC**  **CESS CODES OR MAILBOXES**  **;WHICH CAPRI REMOTE USER DO NOT HAVE, WE HAVE TO NEW DUZ AND CHANGE XMD**  **UZ TO THE NAME OF THE USER**  **;AS A STRING SO THE PROCESS IS STILL LINKED TO THE USER SENDING/TRIGGER**  **ING THE MESSAGE**  **S DVBA0=$G(^DVB(396.3,RIEN,0))**  **S DVBADFN=$P(DVBA0,"^",1),DVBAREQ=$P(DVBA0,"^",4)**  **;S XMDUZ=$P(^VA(200,DVBAREQ,0),"^",1)\_" CAPRI"**  **S XMDUZ="CAPRI "\_$P(^VA(200,.5,0),"^",1)**  **S DVBADT=$$FMTE^XLFDT($P(DVBA0,"^",2))**  **;following call supported by IA 3858**  **S DVBAEA=$P($G(^VA(200,DVBAREQ,.15)),"^",1)**  **S J1=$O(^DVB(396.3,RIEN,6,99999),-1)**  **S J2=$O(^DVB(396.3,RIEN,6,J1,1,99999),-1)**  **S J10=^DVB(396.3,RIEN,6,J1,0),J20=^DVB(396.3,RIEN,6,J1,1,J2,0)**  **S RRRJ=$G(^DVB(396.3,RIEN,6,J1,1,J2,1))**  **S:RRRJ="" RRRJ="None"**  **S RRR=$$EXTERNAL^DILFD(396.34,4,,$P(J10,"^",5))**  **S RRD=$P(J10,"^",6)**  **S:RRD="" RRD="None"**  **S RDT=$$EXTERNAL^DILFD(396.34,.01,,$P(J10,"^",1))**  **S RTO=$$EXTERNAL^DILFD(396.34,.02,,$P(J10,"^",7))**  **S RTF=$$EXTERNAL^DILFD(396.34,3,,$P(J10,"^",4))**  **I DVBAEA="" Q**  **S XMY(DVBAEA)="",DVBASITE=$$SITE^VASITE**  **K J1,J10,J2,J20**  **;**  **Q**  **;**  **MGUSR ; set email addresses from mail group**  **; Supported References:**  **; DBIA #10111: Allows FM read access of ^XMB(3.8,D0,0) using DIC.**  **;**  **N MGN,XMB,DIC,MMG,MDIEN,DVEM,MMUS,ERR**  **S MGN="DVBA C 2507 REROUTE",XMB="DVBA CAPRI REROUTE"**  **S DIC="^XMB(3.8,",DIC(0)="QM",X=MGN D ^DIC**  **I +Y<0 S ERR="INVALID MAIL GROUP NAME" Q**  **S MDIEN=+Y,MMG=0**  **I '$$GOTLOCAL^XMXAPIG(MGN) S ERR="NO ACTIVE LOCAL MEMBERS IN MAIL GROUP**  **" K ^TMP("XMERR",$J) Q**  **F S MMG=$O(^XMB(3.8,MDIEN,1,MMG)) Q:MMG="B" D**  **.S MMUS=$P($G(^XMB(3.8,MDIEN,1,MMG,0)),"^",1)**  **.S DVEM=$P($G(^VA(200,MMUS,.15)),"^",1)**  **.Q:DVEM**  **.S XMY(DVEM)=""**  **.;W !,MMG\_" "\_DVEM**  **Q**  **;**  **END ;**  **K RDT,RRD,RRR,RRRJ,RTF,RTO,X,XMY,XMSUB,XMTEXT,MGN,DIC,DIC(0),J,Y,XMDUZ,**  **XMB,ERR**  **K ^TMP($J,"AMIE1"),DVBADFN,DVBASITE,DVBADT,DVBAEA**  **Q** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBC193P | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6 | | | |
| **Related Options** |  | | | |

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

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DD(396.3 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **DVBC193P ;AJF;Conversion of Request Status Field ; 3/18/16**  **;;2.7;AMIE;\*\*193\*\*;**  **;Post conversion for patch 193**  **;This routine converts the Request Status field to a pointer**  **;IEN in file #396.33**  **N A,A1,A2,RO,DA,X**  **S A=0**  **F S A=$O(^DVB(396.3,A)) Q:A="" D**  **.Q:'$D(^DVB(396.3,A,0))**  **.S A1=$P(^DVB(396.3,A,0),"^",18),RO=$P(^DVB(396.3,A,0),"^",3)**  **.S A2=$S(A1="N":1,A1="P":2,A1="S":3,A1="R":4,A1="C":5,A1="X":6,A1="RX":**  **7,A1="T":8,A1="NT":9,A1="CT":10,1:A1)**  **.S $P(^DVB(396.3,A,0),"^",18)=A2**  **.;Convert "AF" xref**  **.Q:A=""**  **.Q:A1=""**  **.S DA=A,X=A1 X ^DD(396.3,17,1,1,2)**  **.Q:A2=""**  **.S X=A2 X ^DD(396.3,17,1,1,1)**  **Q** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCXFR1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | KILL^DVBCUTIL  ^DVBCXULT  ^DVBCUTL4  ^DVBCXFR2  ^DIC  ^%ZIS |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DIC(4.2 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **DVBCXFR1 ;AJF/ReRoute C&P Request ; 7/18/16 2:14 PM**  **;;2.7;AMIE;\*\*193\*\*;**  **;**  **;**  **EN(RTN,REQDA,PIEN,INUM,RDIV,RR,RD) ;Call from CAPRI**  **; RPC: DVBA CAPRI SEND REROUTE**  **;**  **;ReRoute C&P 2507 Request**  **; RTN = Return value**  **; REQDA = 2507 Request IEN**  **; INUM = Institution IEN**  **; PIEN = Patient IEN**  **; RDIV = Division IEN**  **; RR = Reroute Reason**  **; RD = Reroute Description**  **;**  **I REQDA="" S RTN="0^Missing request number number" Q**  **I INUM="" S RTN="0^Missing Institution number" Q**  **I PIEN="" S RTN="0^Missing Patient IEN" Q**  **I RDIV="" S RTN="0^Missing Routing Location" Q**  **I RR="" S RTN="0^Missing Reroute Reason" Q**  **;**  **I '$D(^DVB(396.3,REQDA,0)) S RTN="0^Not a vadilate request number" Q**  **I '$D(^DIC(4,INUM)) S RTN="0^Not a vadilate Institution IEN" Q**  **;**  **N R0,DFN,DIEN,DOMNAM,DOMNUN,DOMNUM1,EXAMS,EXMNM,EXR,JJ,PNAM**  **N SSN,XMCNT,XMVAR**  **S R0=^DVB(396.3,REQDA,0)**  **K DVBAINSF**  **I $P(R0,U,18)>2 S RTN="0^This request does not have a NEW or PENDING st**  **atus and may not be rerouted." Q**  **;I $P(R0,U,18)'=1,($P(R0,U,18)'=2),($P(R0,U,18)'=12) S RTN="0^This requ**  **est does not have a NEW or PENDING status and may not be rerouted." Q**  **;I $P(R0,U,22)]"" S RTN="0^This request was transferred in and CANNOT b**  **e transferred to any other site " Q**  **;**  **S DFN=$P(R0,U,1)**  **I DFN'=PIEN S RTN="0^Patient IEN passed in does match 2507 Request" Q**  **S PNAM=$P(^DPT(DFN,0),U,1),SSN=$P(^(0),U,9)**  **S:$P(R0,U,10)="E" DVBAINSF=""**  **S DIEN=$P(^DIC(4,INUM,6),U,1)**  **S DOMNUM=$S($P(^DIC(4.2,DIEN,0),U,3)]"":$P(^(0),U,3),1:DIEN)**  **S DOMNAM=$P(^(0),U,1),DOMNUM1=DIEN**  **;I $D(CORR) G DISPLAY**  **;**  **;EXAMS K XEXAMS W @FF,!,"Exam selection",!!!! S EXAMS="",XMCNT=0**  **;F LPCNT=0:0 S LPCNT=$O(XMVAR(LPCNT)) Q:LPCNT="" K XMVAR(LPCNT)**  **;W !!,"Do you want to transfer ALL exams" S %=2 D YN^DICN G:%<0 EXIT**  **;I %=2 W !! G PART**  **;I %=0 W !!,"Enter Y if you want to transfer all exams or N if not.",!!**  **D CONTMES^DVBCUTL4 G EXAMS**  **;W !!!**  **K XEXAMS S EXAMS="",XMCNT=0**  **F JJ=0:0 S JJ=$O(^DVB(396.4,"C",REQDA,JJ)) Q:JJ="" D SET**  **;**  **D INREAS^DVBCXUTL**  **MAILMAN D ^DVBCXFR2**  **;**  **Q**  **;**  **SET ;\*\* EXAMS - Xfr all**  **S EXMNM=$P(^DVB(396.6,$P(^DVB(396.4,JJ,0),U,3),0),U,1)**  **I $P(^DVB(396.4,JJ,0),U,4)["X" S EXR(JJ)=EXMNM\_" is CANCELED and cannot**  **be transferred." Q**  **I $P(^DVB(396.4,JJ,0),U,4)="C" S EXR(JJ)=EXMNM\_" is COMPLETED and canno**  **t be transferred." Q**  **I $P(^DVB(396.4,JJ,0),U,4)="T" S EXR(JJ)=EXMNM\_" has been TRANSFERRED a**  **nd cannot be selected." Q**  **;W !,EXMNM," is OK to transfer.",!!**  **S EXAMS=EXAMS\_$P(^DVB(396.4,JJ,0),U,3)\_U,XEXAMS(JJ)="",XMCNT=XMCNT+1**  **;**  **;\*\* Set XMVAR(XMCNT)=$EXAM AMIE EXAM IFN^INSUFF REASON IFN**  **S XMVAR(XMCNT)="$EXAM "\_$P(^DVB(396.4,JJ,0),U,3)\_U\_$S(+$P(^DVB(396.4,JJ**  **,0),U,11)>0:$P(^DVB(396.94,$P(^DVB(396.4,JJ,0),U,11),0),U,1),1:"")**  **;EXAMS for MailMan msg, XEXAMS sets exam status**  **;XMVAR() add one exam/line to bulletin - Future**  **Q**  **;**  **;**  **EXIT D CLRVAR^DVBCXUTL,KILLVRS^DVBCXUTL,KILL^DVBCUTIL** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCXFR2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBCXFR1 | ^DVBAUTL1  ^DVBCTIL  ^XMD  ^DIE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DG(391  ^DIC(21  ^DIC(4  ^DIC(4.2  ^DIC(5  ^DIC(8  ^DPT  ^DVB(396.3  ^TMP(“DVBCXFR”  ^VA(200  ^XMB(1 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **DVBCXFR2 ;AJF/ReRoute C&P REQUESTS ; 7/25/16 1:55 PM**  **;;2.7;AMIE;\*\*193\*\*;**  **;**  **;Copy of DVBCXFRB**  **;**  **EN ;file 2**  **N DVBSBRCH,DVBDTYPE**  **K ^TMP("DVBCXFR",$J),L S X=^DVB(396.3,REQDA,0),DFN=$P(X,U,1)**  **S X=^DPT(DFN,0),PNAM=$P(X,U,1),DOB=$P(X,U,3),SEX=$P(X,U,2),SSN=$P(X,U,9**  **)**  **S X=$S($D(^DPT(DFN,.11)):^(.11),1:""),ADR1=$P(X,U,1),ADR2=$P(X,U,2),ADR**  **3=$P(X,U,3),CITY=$P(X,U,4),STATE=$P(X,U,5),ZIP=$P(X,U,6),CNTY=$P(X,U,7),ZIP4=$P(**  **X,U,12)**  **S CNTY=$S($D(^DIC(5,+STATE,1,+CNTY,0)):$P(^(0),U,1),1:""),STATE=$S($D(^**  **DIC(5,+STATE,0)):$P(^(0),U,1),1:"")**  **S X=$S($D(^DPT(DFN,.13)):^(.13),1:"") S HOMPHON=$P(X,U,1),BUSPHON=$P(X,**  **U,2)**  **S X=$S($D(^DPT(DFN,.3)):^(.3),1:""),SRVCON=$P(X,U,1),SRVPCT=$P(X,U,2)**  **S X=$S($D(^DPT(DFN,.31)):^(.31),1:""),CNUM=$P(X,U,3)**  **S CFLOC=$$STATION^DVBAUTL1(DFN)**  **S:CFLOC=-1 CFLOC=""**  **S X=$S($D(^DPT(DFN,.32)):^(.32),1:""),PDSRV=$P(X,U,3),PDSRV=$S($D(^DIC(**  **21,+PDSRV,0)):$P(^(0),U,3),1:"")**  **S X=$$SVC^DVBCUTIL(DFN,"I"),SRVEDT=$P(X,U),SRVSDT=$P(X,U,2)**  **S X=$$SVC^DVBCUTIL(DFN,"E"),DVBSBRCH=$P(X,U,3),DVBDTYPE=$P(X,U,4)**  **S X=$S($D(^DPT(DFN,.36)):^(.36),1:""),ELIGCOD=$P(X,U,1),ELIGCOD=$S($D(^**  **DIC(8,+ELIGCOD,0)):$P(^(0),U,9),1:"")**  **S X=$S($D(^DPT(DFN,.361)):^(.361),1:""),ELIGST=$P(X,U,1),ELIGSDT=$P(X,U**  **,2)**  **S X=$S($D(^DPT(DFN,.52)):^(.52),1:""),POWSTAT=$P(X,U,5)**  **S X=$S($D(^DPT(DFN,"VET")):^("VET"),1:""),VETST=$P(X,U,1)**  **S X=$S($D(^DPT(DFN,"TYPE")):^("TYPE"),1:"") S TYPE=$S(X]"":$P(^DG(391,X**  **,0),U,1),1:"")**  **S LINE(1)="$DEM0 "\_PNAM\_U\_DOB\_U\_SEX\_U\_SSN**  **S LINE(2)="$DEM1 "\_ADR1\_U\_ADR2\_U\_ADR3\_U\_CITY\_U\_STATE\_U\_CNTY\_U\_ZIP\_U\_HOM**  **PHON\_U\_BUSPHON\_U\_ZIP4**  **S LINE(3)="$ELIG "\_SRVCON\_U\_SRVPCT\_U\_CFLOC\_U\_CNUM\_U\_PDSRV\_U\_SRVEDT\_U\_SR**  **VSDT\_U\_ELIGCOD\_U\_ELIGST\_U\_ELIGSDT\_U\_POWSTAT\_U\_VETST\_U\_TYPE\_U\_DVBSBRCH\_U\_DVBDTYPE**  **;**  **;file 396.3**  **EN1 S X=^DVB(396.3,REQDA,0),RO=$P(X,U,3),RONAM=$P($G(^DIC(4,+RO,99)),U,1),R**  **EQDT=$P(X,U,2),FEXM=$P(X,U,9),PRIO=$P(X,U,10),OTHDIS=$P(X,U,11),LREXMDT=$P(X,U,2**  **0)**  **S X=$S($D(^DVB(396.3,REQDA,1)):^(1),1:""),CFREQ=$P(X,U,2),OTHDOC=$P(X,U**  **,3),LREXMDT=$P(X,U,7),OTHDIS1=$P(X,U,9),OTHDIS2=$P(X,U,10)**  **S CLTYP=$G(^DVB(396.3,REQDA,9,1,0)),RRDIV=$P($G(^DVB(396.3,REQDA,1)),"^**  **",4),(SPEC,ECF)="",II=0**  **I $D(^DVB(396.3,REQDA,5)) S ECF=$P(^DVB(396.3,REQDA,5),"^",3)**  **I $D(^DVB(396.3,REQDA,8)) F S II=$O(^DVB(396.3,REQDA,8,II)) Q:II=""!(I**  **I="B") D**  **. S SPEC=SPEC\_$G(^DVB(396.3,REQDA,8,II,0))\_"^"**  **S LINE(4)="$REQ0 "\_REQDA\_U\_RO\_U\_PRIO\_U\_CFLOC\_U\_LREXMDT\_U\_CFREQ\_U\_LREXMD**  **T\_U\_RONAM\_U\_RDIV**  **S LINE(5)="$ODIS "\_OTHDIS\_U\_OTHDIS1\_U\_OTHDIS2**  **S EXAMS="$EXAM "\_EXAMS,LINE(6)=EXAMS**  **S RRF=^DD("SITE",1),RRDT=$$NOW^XLFDT()**  **S RRFD=$P(^DIC(4,RRF,6),U,1)**  **; AJF ; Reroute data for MailMan message**  **S LINE(7)="$RDAT "\_REQDA\_U\_PIEN\_U\_RRF\_U\_RR\_U\_RD\_U\_INUM\_U\_RRDT\_U\_CLTYP\_U**  **\_ECF\_U\_RRFD**  **S LINE(8)="$SPEC "\_SPEC**  **;**  **;\*\*NOTE: RONAM is now RO NUMBER**  **LOAD S L=1,^TMP("DVBCXFR",$J,L,0)="$TRANSFER IN",L=L+1**  **F X=1:1 Q:'$D(LINE(X)) S ^TMP("DVBCXFR",$J,L,0)=LINE(X),L=L+1**  **F JI=0:0 S JI=$O(^DVB(396.3,REQDA,2,JI)) Q:JI="" S ^TMP("DVBCXFR",$J,L**  **,0)="$REMK "\_^DVB(396.3,REQDA,2,JI,0),L=L+1**  **S SITE=$P(^XMB(1,1,0),U,1),SITE1=$P(^DIC(4.2,SITE,0),U,1)**  **S USERNM=$P(^VA(200,DUZ,0),U,1),^TMP("DVBCXFR",$J,L,0)="$USER "\_USERNM\_**  **U\_SITE\_U\_SITE1,L=L+1**  **S ^TMP("DVBCXFR",$J,L,0)="$END "**  **N XMNODE**  **S XMNODE=0**  **F X=(L+1):1:(L+XMCNT) DO**  **.S XMNODE=XMNODE+1**  **.S ^TMP("DVBCXFR",$J,X,0)=XMVAR(XMNODE)**  **S ^TMP("DVBCXFR",$J,X+1,0)="$END1 "**  **;**  **SEND K XMZ**  **;S XMY("POSTMASTER@"\_DOMNAM)=DOMNUM**  **N RRIF**  **S RRF=^DD("SITE",1),RRXM=0**  **;Check for reroute within VAMC**  **I RRF'=INUM D**  **.S XMY(DUZ)="",XMY("S.DVBA C PROCESS MAIL MESSAGE@"\_DOMNAM)=DOMNUM,XMSU**  **B="Transfer of C&P Exams",XMTEXT="^TMP(""DVBCXFR"",$J,",XMDUZ=DUZ**  **.D ^XMD**  **.I $D(XMZ) S RTN="1^Transmitted as message # "\_XMZ\_" from this site to**  **"\_DOMNAM**  **.I '$D(XMZ) S RTN="0^Message transmission error! Request WILL NOT be re**  **routed!",RRXM=1**  **;if all ok, update main, sub-file**  **I RRXM=1 D VKILL Q**  **I RRF=INUM S RTN="1^Rerouted to another division within "\_DOMNAM**  **F III=0:0 S III=$O(XEXAMS(III)) Q:III="" D**  **.S DIE="^DVB(396.4,",DA=III,DR=".04///T;62///"\_DOMNAM\_";60////"\_DT\_";61**  **///"\_USERNM**  **.D ^DIE**  **K DIE,DA,DR S DIE="^DVB(396.3,",DA=REQDA,DR="17///14"**  **S CSITE=^DD("SITE",1),MDIV=$P(^DVB(396.15,RDIV,0),"^")**  **I CSITE=RRF&(CSITE=INUM) S DR=DR\_";24////"\_MDIV**  **D ^DIE ;set transfer items**  **;**  **S RRIF=$$UPRR^DVBCUTL8(REQDA,RRDT)**  **S DA=$P(RRIF,"^")**  **S DIE="^DVB(396.3,"\_REQDA\_",6,",DA(1)=REQDA**  **S DR="1////"\_REQDA\_";2////"\_PIEN\_";3////"\_RRF\_";4////"\_RR\_";5////"\_RD**  **S DR=DR\_";.02////"\_INUM\_";8////"\_RRDIV\_";7////"\_DUZ**  **D ^DIE ;set Reroute fields**  **S RRIEN=DA,RRST="N",RRR=""**  **D UPRS^DVBCUTL8(REQDA,RRIEN,RRDT,RRST,RRR) ; Update the status**  **; Send Reroute message to Requestor**  **D SENDMSG^DVBAB1C(REQDA)**  **;**  **; REQDA = 2507 Request IEN**  **; INUM = Institution IEN**  **; PIEN = Patient IEN**  **; DIEN = Division IEN**  **; RR = Reroute Reason**  **; RD = Reroute Description**  **;**  **;**  **;**  **K LINE,DOMNUM,DOMNUM1,^TMP("DVBCXFR",$J),XMDUZ,III,L,JI,JY,XMY,XMZ,XMSU**  **B,XMTEXT,XMDUZ,DIE,DA,DR,TYPE**  **;**  **VKILL ; Kill varables from process**  **;**  **K XMERR,ADR1,ADR2,ADR3,BUSPHON,CFLOC,CFREQ,CITY,CLTYP,CNTY,CNUM**  **K CSITE,DFN,DOB,DOMNAM,ECF,ELIGCOD,ELIGSDT,ELIGST,EXAMS,FEXM**  **K HOMPHON,II,INUM,LREXMDT,MDIV,OTDIS,OTHDIS1,OTHDIS2,OTHDOC,PDSRV**  **K PIEN,PNAM,POWSTAT,PRIO,RD,RDIV,REQDA,REQDT,RR,RONAM,RO,RRDT,RRDIV**  **K RRFD,RRIEN,RRR,RRST,RRXM,SEX,SITE,SITE1,SPEC,SRVCON,SRVEDT,SRVSDT**  **K SSN,STATE,USERNM,VETST,X,XEXAMS,XMCNT,XMVAR,ZIP,ZIP4**  **D KILL^DVBCUTIL,KILL^DVBCUTL2,KILL^DVBCUTL3**  **Q** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCXFRC | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DIC  FILE^DICN  ^DIE  ^DIK  ^DILFD  ^DIQ  ^DVBCUTIL  ^DVBCUTL4  ^XMB  $$PRIM^VASITE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD  ^DG(391  ^DIC(21  ^DIC(4  ^DIC(5  ^DIC(8  ^DPT  ^DVB(396.3  ^DVB(396.4  ^VA(200 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCXFRC ;ALB/GTS-557/THM-PROCESS TRANSFER-IN MAIL MESSAGE ; 10/4/91 9:26 AM  ;;2.7;AMIE;\*\*1,6,18,65,149\*\*;Apr 10, 1995;Build 16  ;Per VHA Directive 2004-038, this routine should not be modified.  REMK S:'$D(CNT) CNT=0 S CNT=CNT+1,REMK(CNT)=XLN  Q  ;  SPEC F II=1:1 S SPEC(II)=$P(XLN,"^",II) Q:SPEC(II)=""  S DR="21////"\_CFREQ\_";23.3////"\_LREXMDT\_";24////"\_DVBCDIV  D ^DIE K DIC,DIE,DD,DO  S CNT=0 I '$D(^DVB(396.3,REQDA,2,0)) S ^(0)="^^0^0^"\_DT\_"^^^^"  F ZI=0:0 S ZI=$O(REMK(ZI)) Q:ZI="" S X=REMK(ZI) S CNT=CNT+1,^DVB(396.3 |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCXFRC ;ALB/GTS-557/THM-PROCESS TRANSFER-IN MAIL MESSAGE ; 10/4/91 9:26 AM  ;;2.7;AMIE;\*\*1,6,18,65,149,**193**\*\*;Apr 10, 1995  ;Per VHA Directive 2004-038, this routine should not be modified.  REMK S:'$D(CNT) CNT=0 S CNT=CNT+1,REMK(CNT)=XLN  Q  ;  **; AJF ; 2507 Reroute fields**  **RDAT S OREQDA=$P(XLN,"^",1),PIEN=$P(XLN,"^",2),RRF=$P(XLN,"^",3)**  **S RR=$P(XLN,"^",4),RD=$P(XLN,"^",5),RRT=$P(XLN,"^",6)**  **S RRDT=$P(XLN,"^",7),CLTY=$P(XLN,"^",8),ECF=$P(XLN,"^",9)**  **S RRFD=$P(XLN,"^",10)**  **Q**  ;  SPEC F II=1:1 S SPEC(II)=$P(XLN,"^",II) Q:SPEC(II)=""  S DR="21////"\_CFREQ\_";23.3////"\_LREXMDT\_";24////"\_DVBCDIV\_";21.1////"\_E  CF  D ^DIE K DIC,DIE,DD,DO  **; AJF ;New Reroute fields**  **S RIEN=REQDA**  **S RRIF=$$UPRR^DVBCUTL8(DA,RRDT)**  **S DA=$P(RRIF,"^")**  **S DIE="^DVB(396.3,"\_RIEN\_",6,",DA(1)=RIEN**  **S DR="1////"\_OREQDA\_";2////"\_PIEN\_";3////"\_RRF\_";4////"\_RR\_";5////"\_RD\_**  **";.02////"\_RRT**  **D ^DIE ;set Reroute fields**  **S RRIEN=DA,RRST="N",RRR=""**  **S RRUP=$$UPRS^DVBCUTL8(RIEN,RRIEN,RRDT,RRST,RRR) ; Update the status**  **;**  **S II=0**  **F S II=$O(SPEC(II)) Q:II="" D**  **.S DIC="^DVB(396.3,"\_RIEN\_",8,",DA(1)=RIEN,DLAYGO=396.3,DIC(0)="L",X=SP EC(II)**  **.D FILE^DICN K DIC**  **;**  **S DIC="^DVB(396.3,"\_RIEN\_",9,",DA(1)=RIEN,DLAYGO=396.3,DIC(0)="L",X=CLT**  **Y**  **D FILE^DICN K DLAYGO,DIC**  **;**  S CNT=0 I '$D(^DVB(396.3,REQDA,2,0)) S ^(0)="^^0^0^"\_DT\_"^^^^"  F ZI=0:0 S ZI=$O(REMK(ZI)) Q:ZI="" S X=REMK(ZI) S CNT=CNT+1,^DVB(396.3 |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCXFRS | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBCXFRC | KILL^DVBCUTIL  DELSER^DVBCUTL4  ^DVBCXFRD  ^XMB  ^DIE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DPT  ^DVB(396.3  ^DVB(396.4  ^DVB(396.6 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCXFRS ;ALB/GTS-557/THM-STUFF C&P TRANSFER RESULTS ; 5/30/91 9:59 AM  ;;2.7;AMIE;\*\*10\*\*;Apr 10, 1995  ;  ROPN S ALLROPN=1,(DIC,DIE)="^DVB(396.3,",DA=REQDA  S DR="11///@;31////y;34///@;17////P"  D ^DIE ;transfers cancelled in error |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCXFRS ;ALB/GTS-557/THM-STUFF C&P TRANSFER RESULTS ; 5/30/91 9:59 AM  ;;2.7;AMIE;\*\*10,**193**\*\*;Apr 10, 1995  ;  ROPN S ALLROPN=1,(DIC,DIE)="^DVB(396.3,",DA=REQDA  S DR="11///@;31////y;34///@;**17////2" ;AJF; Status Conversion**  D ^DIE ;transfers cancelled in error |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCUTL8 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBCCHKR  ^DVBCLKTL  ^DVBCMKL2  ^DVBFXFRC  ^DVBCXFR2  ^DVBCUTL5  ^DVBCSCHD  ^DVBCROPN  ^DVBCREDT  ^DVBCMKL2  ^DVBCLKTL  ^DVBCHS0  ^DVBCCNCL  ^DVBCADEX  ^DVBARSBD  ^DVBABEBD  ^DVBAB70  ^DVBAB6  ^DVBAB5  ^DVBAB2 | FILE^DICN  ^DIE  $$EXTERNAL^DILFD  ^DIR  MSG^DVBAB1C  NOLNK^DVBCLKT2  $$GETDTE^DVBCMKLK  LINKDISP^DVBCUTA1  LNKARY^DVBCUTA3  $$SITE^VASITE  $$NOW^XLFDT |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DD(“SITE”  ^DG(40.8  ^DPT  ^DVB(396.1  ^DVB(396.15  ^DVB(396.3  ^DVB(396.33  ^DVB(396.55  ^DVB(396.95  ^TMP(“DVBC” | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCUTL8 ;ALB/GTS-AMIE C&P APPT LINK FILE MNT RTNS 2 ; 10/20/94 3:30 PM  ;;2.7;AMIE;;Apr 10, 1995  ;  .I $D(DVBAAPST) K DVBAAPT,DVBAAPST  K Y  Q +SELDA |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCUTL8 ;ALB/GTS-AMIE C&P APPT LINK FILE MNT RTNS 2 ; 10/20/94 3:30 PM  ;;2.7;AMIE**;\*\*193\*\*;**Apr 10, 1995  ;  .I $D(DVBAAPST) K DVBAAPT,DVBAAPST  K Y  Q +SELDA  ;  **;AJF; Request Status Conversion**  **RSTAT(RSP) ;\*\*Return Request Status Code from 396.33**  **;RSP - IEN for file 396.33**  **Q:'$D(RSP) ""**  **Q:'+RSP ""**  **Q:'$D(^DVB(396.33,RSP,0)) ""**  **Q $P(^DVB(396.33,RSP,0),"^",2)**  **;**  **;AJF; Request Status Conversion**  **RTSTAT(RSP) ;\*\*Return Status (External) from 396.33**  **;RSP - IEN for file 396.33**  **Q:'$D(RSP) ""**  **Q:'+RSP ""**  **Q:'$D(^DVB(396.33,RSP,0)) ""**  **Q $P(^DVB(396.33,RSP,0),"^",1)**  **;**  **;AJF ; Reroute function**  **REROST(RTN,RSP) ;\*\*Returns 1 if this Request is able to be rerouted**  **;RPC: DVBA CAPRI GET REROUTE**  **;RSP - IEN for file 396.3**  **;RTN - Return value 1 for yes 0 for no**  **Q:'$D(RSP) 0**  **Q:'+RSP 0**  **Q:'$D(^DVB(396.3,RSP,0)) 0**  **N CSITE,RSTA,FSITE**  **S RTN=0**  **S CSITE=+$$SITE^VASITE**  **S FSITE=$S('$D(^DVB(396.3,RSP,6,1,0)):CSITE,1:$P(^DVB(396.3,RSP,6,1,0),**  **"^",4))**  **S RSTA=$P(^DVB(396.3,RSP,0),"^",18)**  **I CSITE=FSITE S:RSTA=1!(RSTA=2)!(RSTA=12) RTN=1**  **S RTN=RTN\_"^"\_CSITE**  **Q**  **;**  **CDIV(RTN,SITE) ;AJF ; Provides list from CAPRI DIVISION EXAM (396.15**  **;RPC DVBA CAPRI GET DIVISION**  **;RTN - Return list of active divisions "^" Division IEN**  **;**  **N CNT,DN,DVP,DV0,FNUM**  **S I=0,RTN(1)="No active CAPRI Divisions"**  **F S I=$O(^DVB(396.15,I)) Q:I="B"!(I="") D**  **. Q:$P($G(^DVB(396.15,I,3)),"^")="Y"**  **. S CNT=$G(CNT)+1,DVP=$P(^DVB(396.15,I,0),"^")**  **. Q:DVP=""**  **. S DV0=$G(^DG(40.8,DVP,0))**  **. S DN=$P(DV0,"^",1),FNUM=$P(DV0,"^",2)**  **. S RTN(CNT)=DN\_" "\_FNUM\_"^"\_I**  **Q**  **;**  **CDIVC(RTN,DIV) ;AJF; Provides comments for GUI**  **; RPC: DVBA CAPRI GET DIV COMMENT**  **; RTN - Return comment**  **; DIV - Division IEN**  **N I**  **S I=0,RTN(1)="No Division comment available "**  **Q:'$D(DIV)**  **Q:'+DIV**  **F S I=$O(^DVB(396.15,DIV,2,I)) Q:I="" D**  **.Q:'$D(^DVB(396.15,DIV,2,I,0))**  **. S RTN(I)=^DVB(396.15,DIV,2,I,0)**  **Q**  **CDIVE(RTN,DIV) ;AJF ; Provides list of active exams**  **; RPC: DVBA CAPRI GET DIV EXAM**  **; RTN - Return exam**  **; DIV - Division IEN**  **N C2,C3,EN,CNT**  **S (C2,CNT)=0,RTN(1)="No exam found"**  **Q:'$D(DIV)**  **Q:'+DIV**  **F S C2=$O(^DVB(396.15,DIV,1,C2)) Q:C2="B"!(CNT=100) D**  **. Q:"DEFAULT "'[$E(^DVB(396.15,DIV,1,C2,0),1,7)**  **. S C3=0**  **. F S C3=$O(^DVB(396.15,DIV,1,C2,3,C3)) Q:C3=""!(CNT=100) D**  **.. Q:'$D(^DVB(396.15,DIV,1,C2,3,C3,0))**  **.. Q:$G(^DVB(396.15,DIV,1,C2,3,C3,2))'="Y"**  **.. S EN=$P(^DVB(396.15,DIV,1,C2,3,C3,0),"^")**  **.. S EN=$$EXTERNAL^DILFD(396.1514,.01,,EN,)**  **.. S CNT=CNT+1,RTN(CNT)=EN**  **Q**  **;**  **ARC(RTN) ;AJF ;7/15/2016 Returns all active Reroute Code**  **; RPC: DVBA CAPRI GET REROUTE CODE**  **; RTN - Return exam**  **;**  **N CT,C1,R0**  **S CT=0**  **F S CT=$O(^DVB(396.55,CT)) Q:CT="B" D**  **. S R0=^DVB(396.55,CT,0)**  **. Q:$P(R0,"^",2)="I"**  **. S C1=$G(C1)+1**  **. S RTN(C1)=CT\_"^"\_$P(R0,"^")**  **Q**  **;**  **RINFO(RTN,RIEN) ;AJF; Returns reroute information for a given 2507 Request**  **;RPC: DVBA CAPRI REROUTE INFO**  **;Input**  **; RIEN: 2507 Request IEN**  **;**  **;Output**  **; REROUTE TO^REROUTE DATE^REROUTE STATUS^STATUS DATE^REROUTED FROM^ RE**  **ROUTE REASON ^ REJECT REASON**  **; ^ 0 for site A/ 1 for site B or C**  **;**  **N RTD,RTF,RTO,RTS,RTSD,RRD,RRD,J1,J2,J10,J20**  **N REJR,RRW1,RRW2,RUSR,RDIV,RTDIV,RFDIV**  **I RIEN="" S RTN="0^Missing 2507 Request IEN" Q**  **I '$D(^DVB(396.3,RIEN,0)) S RTN="0^Not a valid 2507 Request IEN" Q**  **I '$D(^DVB(396.3,RIEN,6,0)) S RTN="0^This 2507 Request has not been Rer**  **outed" Q**  **;**  **S J1=$O(^DVB(396.3,RIEN,6,99999),-1)**  **S J2=$O(^DVB(396.3,RIEN,6,J1,1,99999),-1)**  **I J2="" S RTN="0^This 2507 Request has not been Rerouted" Q**  **S J10=^DVB(396.3,RIEN,6,J1,0),J20=^DVB(396.3,RIEN,6,J1,1,J2,0)**  **S REJR=$G(^DVB(396.3,RIEN,6,J1,1,J2,1))**  **S RTD=$$EXTERNAL^DILFD(396.34,.01,,$P(J10,"^",1))**  **S RTO=$$EXTERNAL^DILFD(396.34,.02,,$P(J10,"^",7))**  **S RTF=$$EXTERNAL^DILFD(396.34,3,,$P(J10,"^",4))**  **S RTSD=$$EXTERNAL^DILFD(396.341,.01,,$P(J20,"^",1))**  **S RTS=$$EXTERNAL^DILFD(396.341,1,,$P(J20,"^",2))**  **S RRR=$$EXTERNAL^DILFD(396.34,4,,$P(J10,"^",5))**  **S RRD=$P(J10,"^",6)**  **S RTDIV=$$EXTERNAL^DILFD(396.3,24,,$P(^DVB(396.3,RIEN,1),"^",4))**  **S RFDIV=$$EXTERNAL^DILFD(396.34,8,,$P(J10,"^",9))**  **;**  **S CSITE=^DD("SITE",1),CRQ=$P(^DVB(396.3,RIEN,0),"^",18),RRW1=0**  **I CSITE=$P(J10,"^",4)&(CSITE=$P(J10,"^",7)) S RRW1=1**  **S RRW2=$S(RRW1:1,CSITE=$P(J10,"^",4):0,1:1)**  **S CST=$S(RRW2=0:0,CRQ=14:1,CRQ=11:1,1:0)**  **;**  **S RTN(1)=RTO\_"^"\_RTD\_"^"\_RTS\_"^"\_RTSD\_"^"\_RTF\_"^"\_RRR\_"^"\_CST\_"^"\_RFDIV**  **\_"^"\_RTDIV**  **S RTN(2)=RRD**  **S RTN(3)=REJR**  **;**  **Q**  **RPRO(RTN,RIEN,RRST,RRR) ; AJF; 7/25/2016; Update Reroute Status**  **;RPC: DVBA CAPRI REROUTE STATUS**  **;Input:**  **; RIEN = 2507 Request IEN**  **; RRST = Reroute status**  **; RRR = Reject Reason**  **;**  **;Output:**  **; RTN = 0 for Failure**  **; 1 for Success**  **;**  **N OSITE,OIEN,DA,DR,DIE,REJM,NSITE,RRUP,J1,J2**  **I RIEN="" S RTN="0^Missing 2507 Request IEN" Q**  **I '$D(^DVB(396.3,RIEN,0)) S RTN="0^Not a valid 2507 Request IEN" Q**  **I '$D(^DVB(396.3,RIEN,6,0)) S RTN="0^This 2507 Request has not been rer**  **outed" Q**  **;**  **S RRR=$G(RRR)**  **S J1=$O(^DVB(396.3,RIEN,6,99999),-1)**  **S J2=$O(^DVB(396.3,RIEN,6,J1,99999),-1)**  **I J2="" S RTN="0^This 2507 Request has not been rerouted" Q**  **S RRIEN=J1,RRDT=$$NOW^XLFDT()**  **S RRUP=$$UPRS(RIEN,RRIEN,RRDT,RRST,RRR)**  **;**  **S R0=^DVB(396.3,RIEN,6,J1,0)**  **S CSITE=^DD("SITE",1),OSITE=$P(R0,"^",4),OIEN=$P(R0,"^",2),NSITE=$P(R0,**  **"^",7)**  **S DIV1=$P(R0,"^",9),DIV2=$P($G(^DVB(396.3,RIEN,1)),"^",4)**  **I CSITE=OSITE,CSITE=NSITE,RRST="R" D**  **. S DIE="^DVB(396.3,"\_RIEN\_",6,",DA=J1,DA(1)=RIEN**  **. S DR="8////"\_DIV2**  **. D ^DIE ;set Reroute fields**  **. K DIE,DA**  **S DA=RIEN,DIE="^DVB(396.3,"**  **; Check to see if this the original site**  **I CSITE=OSITE D**  **. I RRST="A" S DR="6////"\_RRDT\_";17////13" Q**  **. I RRST="R" S DR="17////1",REJM=1**  **I CSITE'=OSITE D**  **. I RRST="A" S DR="17////2" Q**  **. I RRST="R" S DR="6////"\_RRDT\_";17////12"**  **I CSITE=OSITE,CSITE=NSITE D**  **. I RRST="A" S DR="17////2" Q**  **. I RRST="R" S DR="17////1;24////"\_DIV1**  **D ^DIE**  **;**  **; Send Reject Message to DVBA C 2507 Reroute Group**  **D:RRST="R" MSG^DVBAB1C(RIEN)**  **;**  **I CSITE=OSITE S RTN="1^Reroute status updated" Q**  **;**  **S OIEN=$P(R0,"^",2)**  **S RTN="1^Reroute status updated^"\_OSITE\_"^"\_OIEN**  **;**  **Q**  **;**  **;**  **UPRR(RIEN,RRDT) ;AJF ; 7/30/2016; Update Reroute information**  **;create Reroute entry for 2507 Request in sub-file 396.33**  **N DIC,X,Y,DA,DTOUT,DUOUT,DLAYGO,DO**  **S DIC="^DVB(396.3,"\_RIEN\_",6,",DA(1)=RIEN**  **S DIC(0)="L",DLAYGO=396.3**  **S X=RRDT ;.01 2507 REQUEST REROUTE DATE**  **D FILE^DICN K DLAYGO**  **;**  **;**  **Q Y\_"^"\_RRDT**  **;**  **UPRS(RIEN,RRIEN,RRDT,RRST,RRR) ; Update the status**  **N DIC,X,Y,DA,DO,DTOUT,DUOUT,DLAYGO**  **S RRR=$G(RRR)**  **S DIC="^DVB(396.3,"\_RIEN\_",6,"\_RRIEN\_",1,"**  **S DA(1)=RIEN,DA(2)=RRIEN**  **S DIC(0)="FL",DLAYGO=396.3**  **S X=RRDT ;.01 2507 REQUEST REROUTE DATE**  **S DIC("DR")="1////"\_RRST\_";2////"\_RRR**  **D FILE^DICN**  **S R2=Y**  **Q Y** |

Table 15 (Grouping): Routines

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCUTL5 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^SDECCAP  ^SDEC58  ^DVBCCNNS  ^DVBCMKLK  ^DVBCLOG2  ^DVBCSDEV  ^DVBCLKTL | ^DIC  ^DIR  $$RSTAT^DVBCUTL8 |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DPT  ^DVB(396.3  ^DVB(396.4  ^TMP(“DBVC”  ^TMP(“SDEVT” | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCUTL5 ;ALB/GTS-AMIE C&P APPT LINK USER SEL RTNS ; 10/20/94 1:00 PM  ;;2.7;AMIE;;Apr 10, 1995  ;  ;\*\* exam on current 2507, Set ^TMP  F S DVBADA=$O(^DVB(396.3,"B",DVBADFN,DVBADA)) Q:DVBADA="" DO  .I $P(^DVB(396.3,DVBADA,0),U,18)=DVBASTAT DO  ..S DVBAOPEN=$$OPENCHK(DVBADA) I +DVBAOPEN'>0 K DVBAOPEN  F S REQDA=$O(^DVB(396.3,"B",DVBADFN,REQDA)) Q:REQDA="" DO  .I +$P(^DVB(396.3,REQDA,0),U,2)>0,($P(^DVB(396.3,REQDA,0),U,18)'="N") D  O  ..I $P(^DVB(396.3,REQDA,0),U,18)'="" DO  ...S REQDT=$P(^DVB(396.3,REQDA,0),"^",2),DVBACNT=DVBACNT+1  ...S ^TMP("DVBC",$J,9999999.999999-REQDT,REQDT,REQDA)=""  Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCUTL5 ;ALB/GTS-AMIE C&P APPT LINK USER SEL RTNS ; 10/20/94 1:00 PM  ;;2.7;AMIE**;\*\*193\*\*;**Apr 10, 1995  ;  ;\*\* exam on current 2507, Set ^TMP  F S DVBADA=$O(^DVB(396.3,"B",DVBADFN,DVBADA)) Q:DVBADA="" DO  **.;AJF;Request Status conversion ;**  **.I $$RSTAT^DVBCUTL8($P(^DVB(396.3,DVBADA,0),U,18))=DVBASTAT DO**  ..S DVBAOPEN=$$OPENCHK(DVBADA) I +DVBAOPEN'>0 K DVBAOPEN  F S REQDA=$O(^DVB(396.3,"B",DVBADFN,REQDA)) Q:REQDA="" DO  .S REQ=$G(^DVB(396.3,REQDA,0))  **.;AJF;Request Status conversion ;changed "N" to 1**  **.I +$P(REQ,U,2)>0,($P(REQ,U,18)'=1) DO**  **..I $P(REQ,U,18)'="" DO**  **...S REQDT=$P(REQ,"^",2),DVBACNT=DVBACNT+1**  ...S ^TMP("DVBC",$J,9999999.999999-REQDT,REQDT,REQDA)=""  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCUTL3 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBAB96  ^DVBCCHKR  ^DVBCIUTL  ^DVBCPRNT  ^DVBCREQ1  ^DVBCUTL2 | ^DVBAUTL4  ^DVBCREQ1  ^DIR  ^DIWP  ^DIWW |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(31  ^DIC(4.2  ^DPT  ^DVB(396.4  ^DVB(396.5  ^DVB(396.6  ^DVB(396.94  ^TMP(“DVBC”  ^TMP(“DVBCLAB”  ^TMP($J | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCUTL3 ;ALB/GTS-557/THM-DVBCUTL2, CONTINUED ; 5/17/91 11:35 AM  ;;2.7;AMIE;\*\*132\*\*;Apr 10, 1995;Build 2  ;  K HD91,CNUM,DFN,DX,DXCOD,DXNUM,I,LINE,DVBCRALC,PCT,SC,SSN,CANCBY,CANCDT  ,CANCREM,DATRETN,EXST,FA,OWNDOM,REMK,TSTA1,XMFG,YY,ZG,ZH,DVBCAO,XCN,MANUAL  K DVBCTYPE,HD7,HD8,HD9,HD91,RDATE1,TPRT,XJ,RONAM,PNAM,^TMP($J)  Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCUTL3 ;ALB/GTS-557/THM-DVBCUTL2, CONTINUED ; 5/17/91 11:35 AM  ;;2.7;AMIE;\*\*132,**193\***\*;Apr 10, 1995;Build 2  ;  K HD91,CNUM,DFN,DX,DXCOD,DXNUM,I,LINE,DVBCRALC,PCT,SC,SSN,CANCBY,CANCDT  ,CANCREM,DATRETN,EXST,FA,OWNDOM,REMK,TSTA1,XMFG,YY,ZG,ZH,DVBCAO,XCN,MANUAL  K DVBCTYPE,HD7,HD8,HD9,HD91,RDATE1,TPRT,XJ,RONAM,PNAM,^TMP($J)  **K ZTQUEUED,ZIP4,ZIP,Z1,ZH,XRTN,XMVAR,XMRG,XMREC,XMER,XMCNT,XD,WRKSHT,VE**  **TST**  **K USER,USERNM**  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCTRN1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

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

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DVB(396.4  ^DVB(396.6 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCTRN1 ;ALB/JLU;AMIE link routine;12/21/92  ;;2.7;AMIE;\*\*12\*\*;Apr 10, 1995  NSFT K DVBCHK  OREQ(DVBX) ;  ;This function will check to see if there is an open request.  ;DVBCX should contain the REQUEST STATUS.  ;  I $S(DVBX'="P"&(DVBX'="S")&(DVBX'="T")&(DVBX'="NT"):1,1:0) Q 0  Q 1 |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCTRN1 ;ALB/JLU;AMIE link routine;12/21/92  ;;2.7;AMIE;\*\*12,**193**\*\*;Apr 10, 1995  NSFT K DVBCHK  OREQ(DVBX) ;  ;This function will check to see if there is an open request.  ;DVBCX should contain the REQUEST STATUS.  ;  **;AJF; 2507 Request Status Conversion**  **S DVBX=$$RSTAT(DVBX)**  I $S(DVBX'="P"&(DVBX'="S")&(DVBX'="T")&(DVBX'="NT"):1,1:0) Q 0  Q 1 |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCSCHD | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCUTIL  ^SDM  ^XLFDT  ^%ZIS  ^DIC  ^DICN  ^DIE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DVB(396.4  ^DVB(396.6  ^XUSEC | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCSCHD ;ALB/GTS-557/THM-SCHEDULE C&P EXAMS ; 9/23/91 9:54 AM  ;;2.7;AMIE;\*\*17\*\*;Apr 10, 1995  ;  D GO1 I '$D(TFIND) W !!,\*7,"This request has been completely transferre  d to another site.",!,"Scheduling will not be allowed.",!! H 3 G EN  S STAT=$P(^DVB(396.3,+Y,0),U,18) D STATCHK I $D(NCN) G EN  S DTSCHED=$P(^DVB(396.3,+Y,0),U,6) I DTSCHED]"" W !!,\*7,"Scheduling has  been completed for this request as of ",$$FMTE^XLFDT(DTSCHED,"5DZ")\_".",! |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCSCHD ;ALB/GTS-557/THM-SCHEDULE C&P EXAMS ; 9/23/91 9:54 AM  ;;2.7;AMIE;\*\*17,193\*\*;Apr 10, 1995  ;  D GO1 I '$D(TFIND) W !!,\*7,"This request has been completely transferre  d to another site.",!,"Scheduling will not be allowed.",!! H 3 G EN  **;AJF;Conversion of Request Status field**  **S STAT=$P(^DVB(396.3,REQDA,0),U,18),STAT=$$RSTAT^DVBCUTL8(STAT) D STATCHK I $D(NCN) G EN**  S DTSCHED=$P(^DVB(396.3,+Y,0),U,6) I DTSCHED]"" W !!,\*7,"Scheduling has  been completed for this request as of ",$$FMTE^XLFDT(DTSCHED,"5DZ")\_".",! |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCROPN | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCUTIL  ^DVBAUTL4  ^DVBCROP1  ^DVBCUTL4  ^DVBCXFRE  ^%ZIS  ^XMB  ^DIC  ^DIE  ^DICN |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(4.2  ^DD(“DD”  ^DPT  ^DVB(396.3  ^DVB(396.4  ^DVB(396.6  ^TMP($J  ^VA(200  ^XUSEC | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCROPN ;ALB/GTS-557/THM-REOPEN REQUEST/SELECTED EXAMS ; 9/22/91 4:54 PM  ;;2.7;AMIE;\*\*42\*\*;Apr 10, 1995  I $D(DUZ)#2=0 W \*7,!!,"Your user number (DUZ) is invalid !",!! H 3 G EX  IT  LOOK D KILL W @FF,!?(IOM-$L(HD)\2),HD,!?(IOM-$L(HD2)\2),HD2,!!  S DIC("W")="D DICW^DVBCUTIL" S DIC="^DVB(396.3,",DIC(0)="AEQM",DIC("A")  ="Select VETERAN: " D ^DIC G:X=""!(X=U) EXIT I +Y<0 W \*7," ???" G LOOK  S (REQDA,DA(1))=+Y,STAT=$P(^DVB(396.3,DA(1),0),U,18),DFN=$P(Y,U,2)  I STAT="C"!(STAT["X")!(STAT="R")&(SUPER=0) W !!,\*7,"Status prohibits ac  tivity except by supervisors.",!! H 3 G EN  .  .  G:$D(NOTRPT) LOOK  S STAT=$P(^DVB(396.3,DA(1),0),U,18) D STATCHK G:$D(NCN) LOOK  ;  ROPN W !!,"Do you want to reopen the ENTIRE request" S %=2 D YN^DICN G:$D(DT  OUT)!(%<0) EXIT G:%=1 ALL  I $D(Y) W \*7,!,"Reopen error on ",EXMNM," exam !",! H 2  Q  NOTIFY S X=$P(^DVB(396.3,DA(1),0),U,18) I X'["X"&(X'="")&(X'="C") W !!,"Entire  exam is now REOPENED.",!! H 1  I X["X"!(X="")!(X="C") W \*7,!!,"Reopen error !",!! H 3 S OUT=1 K X Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCROPN ;ALB/GTS-557/THM-REOPEN REQUEST/SELECTED EXAMS ; 9/22/91 4:54 PM  ;;2.7;AMIE;\*\*42,193\*\*;Apr 10, 1995  I $D(DUZ)#2=0 W \*7,!!,"Your user number (DUZ) is invalid !",!! H 3 G EX  IT  LOOK D KILL W @FF,!?(IOM-$L(HD)\2),HD,!?(IOM-$L(HD2)\2),HD2,!!  S DIC("W")="D DICW^DVBCUTIL" S DIC="^DVB(396.3,",DIC(0)="AEQM",DIC("A")  ="Select VETERAN: " D ^DIC G:X=""!(X=U) EXIT I +Y<0 W \*7," ???" G LOOK  **;AJF;Request Status conversion**  **S (REQDA,DA(1))=+Y,STAT=$$RSTAT^DVBCUTL8($P(^DVB(396.3,DA(1),0),U,18)),**  **DFN=$P(Y,U,2)**  I STAT="C"!(STAT["X")!(STAT="R")&(SUPER=0) W !!,\*7,"Status prohibits ac  tivity except by supervisors.",!! H 3 G EN  .  .  G:$D(NOTRPT) LOOK  **;AJF;Request Status conversion**  **S STAT=$$RSTAT^DVBCUTL8($P(^DVB(396.3,DA(1),0),U,18)) D STATCHK G:$D(NC**  **N) LOOK**  ;  ROPN W !!,"Do you want to reopen the ENTIRE request" S %=2 D YN^DICN G:$D(DT  OUT)!(%<0) EXIT G:%=1 ALL  .  .  I $D(Y) W \*7,!,"Reopen error on ",EXMNM," exam !",! H 2  Q  **;AJF;Request Status conversion**  **NOTIFY S X=$$RSTAT^DVBCUTL8($P(^DVB(396.3,DA(1),0),U,18)) I X'["X"&(X'="")&(X'**  **="C")** W !!,"Entire exam is now REOPENED.",!! H 1  I X["X"!(X="")!(X="C") W \*7,!!,"Reopen error !",!! H 3 S OUT=1 K X Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCREQP | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCREQ1  ^DVBCREQ2  ^DVBCREQ3  ^DVBCUTIL  ^DVBCUTIL4  ^%DT  ^DIC  ^DIE  ^%ZIS  ^%ZTLOAD |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DG(40.8  ^DVB(396.3  ^TMP($J | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCREQP ;ALB/GTS-557/THM-PRINT NEW REQUESTS ; 6/27/91 9:36 AM  ;;2.7;AMIE;;Apr 10, 1995  S DDNS ="" G EN  .  .  S DA=DA(1),DIE="^DVB(396.3,",DR="17////P" I STAT="N"!(STAT="NT") D ^DIE  SET S DA=DA(1),DR="4///NOW",(DIC,DIE)="^DVB(396.3,"  I $P(^DVB(396.3,DA,0),U,5)="" D ^DIE |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCREQP ;ALB/GTS-557/THM-PRINT NEW REQUESTS ; 6/27/91 9:36 AM  ;;2.7;AMIE;\*\***193**\*\*;;Apr 10, 1995  S DDNS ="" G EN  .  ,  S DA=DA(1),DIE="^DVB(396.3,",DR="17////2" I STAT="N"!(STAT="NT")!(STAT=  "RN") D ^DIE  ;**AJF ; Reroute request; 072016**  **I STAT="RN" S DA=DA(1),DIE="^DVB(396.3,",DR="17////14" D ^DIE**  SET S DA=DA(1),DR="4///NOW",(DIC,DIE)="^DVB(396.3," |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCREDT | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCUTIL  ^DVBCLOG2  ^DVBCUTA2  ^DVBAULT4  ^DVBCUTIL4  ^DIR  ^DIC  ^DIE  ^%ZIS |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DVB(396.3  ^DVB(396.4  ^TMP(“DVBCEDIT”  ^TMP($J | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCREDT ;ALB/GTS-557/THM-EDIT STATIC C&P INFO ; 11/20/90 6:29 AM  ;;2.7;AMIE;;Apr 10, 1995  ;  S STAT=$P(^DVB(396.3,DA,0),U,18) I STAT'="N"&(STAT'="P") W !!,"The stat  us of this request is not NEW or PENDING, REPORTED.",!,"It cannot, therefore, be  modified.",\*7,!! S DVBCMOD=1 G CON |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCREDT ;ALB/GTS-557/THM-EDIT STATIC C&P INFO ; 11/20/90 6:29 AM  ;;2.7;AMIE;\*\*193\*\*;Apr 10, 1995  ;  S STAT=$P(^DVB(396.3,DA,0),U,18)  **;AJF; Request Status Conversion**  **S STAT=$$RSTAT^DVBCUTL8(STAT)**  **I STAT'="N"&(STAT'="P")&(STAT'="NR") W !!,"The status of this request i**  **s not NEW or NEW RE-ROUTED or PENDING, REPORTED.",!,"It cannot, therefore, be mo**  **dified.",\*7,!! S DVBCMOD=1 G CON** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCPND1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCUTL2  ^DVBCUTIL  ^XLFDT  ^XUWORKDY  ^%DTC |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(40.8  ^DIC(4  ^DIC(4.2  ^DPT  ^DVB(396.1  ^DVB(396.3  ^VA(200 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCPND1 ;ALB/GTS-557/THM,SBW-2507 PENDING REQUESTS, PART 2 ; 3/MAY/2011  ;;2.7;AMIE;\*\*17,168\*\*;Apr 10, 1995;Build 3  ;  SORT S STAT=$P(^DVB(396.3,REQDA,0),U,18) Q:"^R^X^RX^C^CT^"[(U\_STAT\_U) I DVB  CSORT="S" Q:STAT'=RSTAT&(RSTAT'="A")  S PNAM=$S($D(^DPT(DFN,0)):$P(^(0),U,1),1:"Unknown"),ROUT=$S($D(^DVB(396  S STATUS="Unknown" W !?10,"Division: "\_ADIV,!?12,"Status: " S XX=$P(^DV  B(396.3,DA(1),0),U,18),STATUS=$S(XX="N":"New",XX="P":"Pending, reported",XX="S":  "Pending, scheduled",XX="R":"Released to RO, not printed",1:"") |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCPND1 ;ALB/GTS-557/THM,SBW-2507 PENDING REQUESTS, PART 2 ; 3/MAY/2011  ;;2.7;AMIE;\*\*17,168,**193\*\*;**Apr 10, 1995  ;  **SORT ; AJF;Request Status Conversion**  **S ST1=$P(^DVB(396.3,REQDA,0),U,18),STAT=$$RSTAT^DVBCUTL8(ST1)**  **Q:"^R^X^RX^C^CT^RR^RA^"[(U\_STAT\_U)**  **I DVBCSORT="S" Q:STAT'=RSTAT&(RSTAT'="A")**  S PNAM=$S($D(^DPT(DFN,0)):$P(^(0),U,1),1:"Unknown"),ROUT=$S($D(^DVB(396  .3,REQDA,1)):$P(^(1),U,4),1:0),RDATE=$P(^(0),U,5)  ;AJF;Conversion  W !?10,"Division: "\_ADIV,!?12,"Status: " S XX=$P(^DVB(396.3,DA(1),0),U,18)  **S STATUS=$$RTSTAT^DVBCUTL8(XX)**  **S XX=$$RSTAT^DVBCUTL8(XX)**  W STATUS,!!,PNAM,?49,"SSN: ",SSN,!?44,"Claim no: ",CNUM,!?40,"Request d  ate: ",$$FMTE^XLFDT(RDATE1,"5DZ"),!?40,"Elapsed days: ",EDAYS,!! |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCHS0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^GMTSAMIE | ^DVBCHS1  ^DVBCHS2 |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DPT  ^DVB(396.3  ^DVB(396.4 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCHS0 ;ALB/JRP - C & P EXTRACT FOR HEALTH SUMMARY;11-JAN-95  ;;2.7;AMIE;;Apr 10, 1995  HSCP(PATPTR,INVBEGDT,INVENDDT,OUTCODE,ARRAY) ;MAIN ENTRY POINT  ..S TMP=$P($G(^DVB(396.3,TMP,0)),"^",18)  ..Q:((TMP'="C")&(TMP'="R")&(TMP'="CT")) |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCHS0 ;ALB/JRP - C & P EXTRACT FOR HEALTH SUMMARY;11-JAN-95  ;;2.7;AMIE;\*\***193**\*\*;Apr 10, 1995  HSCP(PATPTR,INVBEGDT,INVENDDT,OUTCODE,ARRAY) ;MAIN ENTRY POINT    ..S TMP=$P($G(^DVB(396.3,TMP,0)),"^",18)  **..;AJF ; 2507 Request Status Conversion**  **..S TMP=$$RSTAT^DVBCUTL8(TMP)**  ..Q:((TMP'="C")&(TMP'="R")&(TMP'="CT")) |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVFCHLR | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCHLUT  ^VADPT  ^XMB  ^HLFNC  ^HLTRANS  ^DIE  ^%DTC |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DVB(396.3  ^DVB(396.4  ^HL | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCHLR ;ALB/JLU-Processes the results from the ORU ;1/28/93  ;;2.7;AMIE;\*\*9\*\*;Apr 10, 1995  ;  S DIE="^DVB(396.3,",DA=DVBCURQ  S DR="11///NOW;17////T"  D ^DIE |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCHLR ;ALB/JLU-Processes the results from the ORU ;1/28/93  ;;2.7;AMIE;\*\*9,**193\*\*;**Apr 10, 1995  ;  S DIE="^DVB(396.3,",DA=DVBCURQ  ;AJF;Request Status Conversion  S DR="11///**NOW;17////8"**  D ^DIE |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCCNCL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBAB96  ^DVBCCNCL  ^DVBCUTL2 | ^DVBCBULL  ^DVBCCNCL  ^DVBCXFRE  ^DIC  ^DICN  ^DIE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(4.2  ^DVB(396.3  ^DVB(396.4  ^DVB(396.6 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCCNC1 ;ALB ISC/THM-CANCEL ENTIRE REQUEST ; 9/22/91 4:14 PM  ;;2.7;AMIE;\*\*194\*\*;Apr 10, 1995;Build 5  ;  NOTIFY S X=$P(^DVB(396.3,DA,0),U,18) I X="RX"!(X="X") W !!,"Entire exam is now  CANCELLED.",!! H 1 Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCCNC1 ;ALB ISC/THM-CANCEL ENTIRE REQUEST ; 9/22/91 4:14 PM  ;;2.7;AMIE;\*\*194,193\*\*;Apr 10, 1995;Build 5  ;  **NOTIFY S X=$P(^DVB(396.3,DA,0),U,18)**  **;AJF; Request Status Conversion**  **S X=$$RSTAT^DVBCUTL8(X)** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCCNCL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBCCNC1 | ^DVBCCNC1  ^DVBCUTL2  ^DVBCUTL4  ^%ZIS |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DPT  ^DVB(396.3  ^DVB(396.4  ^DVB(396.6  ^TMP | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCCNCL ;ALB/GTS - 557/THM-2507 CANCEL REQUESTS, EXAMS ; 9/23/91 9:25 AM  ;;2.7;AMIE;\*\*102,184,194\*\*;Apr 10, 1995;Build 5  ;  LOOK D KILL W @FF,!?(IOM-$L(HD)\2),HD,!?(IOM-$L(HD2)\2),HD2,!! S DIC("W")="D  DICW^DVBCUTIL" S DIC="^DVB(396.3,",DIC(0)="AEQM",DIC("A")="Select VETERAN: " D  ^DIC G:X=""!(X=U) EXIT I +Y<0 W \*7," ???" G LOOK  S DA(1)=+Y,DFN=$P(Y,U,2),STAT=$P(^DVB(396.3,DA(1),0),U,18) D STATCHK G:  $D(NCN) LOOK S REQDT=$P(^DVB(396.3,DA(1),0),U,2)  S EXMPTR=+Y,EXMNM=$P(^DVB(396.4,+Y,0),U,3)  S EXMNM=$S($D(^DVB(396.6,EXMNM,0)):$P(^(0),U,1),1:"Unknown exam")  S STAT=$P(^TMP($J,EXMNM),U,1) D STATCHK G:$D(NCN) DATA  D CNCLCHK G:NOFND=0 DATA G:$D(OUT) EXIT |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCCNCL ;ALB/GTS - 557/THM-2507 CANCEL REQUESTS, EXAMS ; 9/23/91 9:25 AM  ;;2.7;AMIE;\*\*102,184,193,194\*\*;Apr 10, 1995;Build 5  ;  LOOK D KILL W @FF,!?(IOM-$L(HD)\2),HD,!?(IOM-$L(HD2)\2),HD2,!! S DIC("W")="D  DICW^DVBCUTIL" S DIC="^DVB(396.3,",DIC(0)="AEQM",DIC("A")="Select VETERAN: " D  ^DIC G:X=""!(X=U) EXIT I +Y<0 W \*7," ???" G LOOK  S DA(1)=+Y,DFN=$P(Y,U,2),STAT=$P(^DVB(396.3,DA(1),0),U,18)  **;AJF; Request Status Conversion**  **S STAT=$$RSTAT^DVBCUTL8(STAT)**  S EXMPTR=+Y,EXMNM=$P(^DVB(396.4,+Y,0),U,3)  S EXMNM=$S($D(^DVB(396.6,EXMNM,0)):$P(^(0),U,1),1:"Unknown exam")  S STAT=$P(^TMP($J,EXMNM),U,1)  **;AJF; Request Status Conversion**  **S STAT=$$RSTAT^DVBCUTL8(STAT) W !,"2"\_STAT** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBCADEX | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVNBCEEXM  ^DVBCUTIL  ^DVBCADE2  ^%ZIS  ^DIC  ^DIQ |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DPT  ^DVB(396.3  ^TMP | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBCADEX ;ALB/GTS - 557/THM-ADD C&P EXAMS TO REQUESTS, PART 1 ; 6/28/91 9:32 AM  ;;2.7;AMIE;\*\*184\*\*;Apr 10, 1995;Build 10  ;  S STAT=$P(X,U,18) K NCN  F DTB="X","RX","T","C","R","CT","NT" I STAT=DTB S NCN=1 Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBCADEX ;ALB/GTS - 557/THM-ADD C&P EXAMS TO REQUESTS, PART 1 ; 6/28/91 9:32 AM  ;;2.7;AMIE;\*\*184,193\*\*;Apr 10, 1995  ;  **;AJF;Request Status conversion**  **S STAT=$$RSTAT^DVBCUTL8($P(X,U,18)) K NCN**  F DTB="X","RX","T","C","R","CT","NT" I STAT=DTB S NCN=1 Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBAB2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DBAB3 | ^DVBCAMR1  ^DVBCAMR2  ^DVBCUTA1  ^DVBCUTL2  ^%DT  ^%DTC  ^XMD |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DIC(4  ^DVB(396.3  ^VA(200 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBAB2 ;ALB/KLB - CAPRI RO AMIS REPORT CONT. ;05/01/00  ;;2.7;AMIE;\*\*35,42,149,184\*\*;Apr 10, 1995;Build 10  ;Per VHA Directive 2004-038, this routine should not be modified.  ;  S:(DVBAEXMP']"") DVBAPREXM="ALL" ;identifier for totals  I DTREL'<BDATE,DTREL'>EDATE D DAY30  I DTRPT'<BDATE,DTRPT'>EDATE S TOT(DVBAPREXM,"SENT")=TOT(DVBAPREXM,"SENT  ")+1  I DTRPT'<BDATE,DTRPT'>EDATE,RQSTAT'["X" S X1=$S(DTSCHEDC]"":DTSCHEDC,1:  DVBCNOW),X2=DTRPT D ^%DTC I X>3 S TOT(DVBAPREXM,"3DAYSCH")=TOT(DVBAPREXM,"3DAYSC  H")+1 |

| Modified Logic (Changes are in bold) |
| --- |
| DVBAB2 ;ALB/KLB - CAPRI RO AMIS REPORT CONT. ;05/01/00  ;;2.7;AMIE;\*\*35,42,149,184,193\*\*;Apr 10, 1995  ;Per VHA Directive 2004-038, this routine should not be modified.  ;  S:(DVBAEXMP']"") DVBAPREXM="ALL" ;identifier for totals  **;AJF;Request Status conversion ;**  **S RQSTAT=$$RSTAT^DVBCUTL8(REQDA)** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBAB5 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ^DVBAB6 | ^XUWORKDY  ^%DTC |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(4  ^DPT  ^DVB(396.3 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBAB5 ;ALB/KLB - CAPRI PENDING REQUESTS CONT ;08/01/00  ;;2.7;AMIE;\*\*35,185\*\*;Apr 10, 1995;Build 18  ;  SORT S STAT=$P(^DVB(396.3,REQDA,0),U,18) Q:"^R^X^RX^C^CT^"[(U\_STAT\_U) I DVB  CSORT="S" Q:STAT'=RSTAT&(RSTAT'="A") |

| Modified Logic (Changes are in bold) |
| --- |
| DVBAB5 ;ALB/KLB - CAPRI PENDING REQUESTS CONT ;08/01/00  ;;2.7;AMIE;\*\*35,185,193\*\*;Apr 10, 1995  ;  **SORT ; AJF;Request Status Conversion**  **N ROUT**  **S STAT=$P(^DVB(396.3,REQDA,0),U,18)**  **Q:STAT=""**  **S STAT=$$RSTAT^DVBCUTL8(STAT)**  Q:"^R^X^RX^C^CT^RR^RA^"[(U\_STAT\_U) I DVBCSORT="S" Q:STAT'=RSTAT&(RSTAT  '="A") |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBAB6 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBAB5  ^XLFDT  ^VADPT  ^DIQ  ^DILFD |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DG(40.8  ^DIC(4  ^DIC(4.2  ^DPT  ^DVB(396.3  ^DVB(396.4  ^DVB(396.5  ^DVB(396.6  ^VA(200 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBAB6 ;ALB/DJS - CAPRI PENDING 2507 REQUEST REPORT ; 02-27-2013  ;;2.7;AMIE;\*\*35,90,108,168,185,190,192\*\*;Apr 10, 1995;Build 15  ;  D ELAPSED^DVBAB5  S STATUS="Unknown",XX=$P(^DVB(396.3,DA(1),0),U,18),STATUS=$S(XX="N":"Ne  w",XX="P":"Pending, reported",XX="S":"Pending, scheduled",XX="R":"Released to RO  , not printed",1:"") |

| Modified Logic (Changes are in bold) |
| --- |
| DVBAB6 ;ALB/DJS - CAPRI PENDING 2507 REQUEST REPORT ; 02-27-2013  ;;2.7;AMIE;\*\*35,90,108,168,185,190,192,193\*\*;Apr 10, 1995;Build 15  ;  D ELAPSED^DVBAB5  **;AJF;Request Status conversion Reports ;**  **S STATUS="Unknown",XX=$$RSTAT^DVBCUTL8($P(^DVB(396.3,DA(1),0),U,18))**  S STATUS=$$EXTERNAL^DILFD(396.3,17,,$P(^DVB(396.3,DA(1),0),U,18)) |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBAB70 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DVBCUTIL  ^DVBAB97  ^DVBAB96  ^DVBAB68  ^XLFDT  ^VASITE |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DD(“DD”  ^DIC(31  ^DPT  ^DVB(396.3  ^DVB(396.4  ^VA(200 | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBAB70 ;ALB/SPH - CAPRI C&P EXAM INQUIRY ;09/28/2009  ;;2.7;AMIE;\*\*35,42,57,136,143,149\*\*;Apr 10, 1995;Build 16  ;Per VHA Directive 2004-038, this routine should not be modified.  S (XSTAT,STAT)=$P(^DVB(396.3,REQDA,0),U,18)  S STAT=$S(XSTAT="N":"New",XSTAT="P":"Pending, reported to MAS",XSTAT="T  ":"Transcribed",XSTAT="S":"Scheduled",XSTAT="R":"Released, not printed",XSTAT="C  ":"Completed",XSTAT="CT":"Completed, transferred out",XSTAT="NT":"New, transferr  ed in",1:"") |

| Modified Logic (Changes are in bold) |
| --- |
| DVBAB70 ;ALB/SPH - CAPRI C&P EXAM INQUIRY ;09/28/2009  ;;2.7;AMIE;\*\*35,42,57,136,143,149,**193**\*\*;Apr 10, 1995  ;Per VHA Directive 2004-038, this routine should not be modified.  ;  **S (XSTAT,STAT)=$P(^DVB(396.3,REQDA,0),U,18)**  **S XSTAT=$$RSTAT^DVBCUTL8(STAT)**  **S STAT=$$RTSTAT^DVBCUTL8(STAT)** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBACRVA | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

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

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | GETS^DIQ  $$RTSTAT^DVBCUTL8  DEM^VADPT  $$FMTE^XLFDT | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **DVBACRVA ;SLC/GRE - ; 01/08/2016  ;;5.01;CAPRI;\*\*193\*\*;AUG 30,2016;Build 1  ;  ;Public, Supported ICRs  ; #2056 - Fileman API - $$GET1^DIQ  ;  Q  ; GETFACNM(RESULT,SORTBY) ; Extract records from file and sort by State Code  K ^TMP($J)  N II,FACNAME,STATE,ST,STATION,STATX,FIEN,FACTYPE,R,F1,F11,F2,F2N,VAMCNAME  S (FACNAME,STATE,ST,STATION,STATX,FIEN,FACTYPE,F1,F11,F2,F2N,VAMCNAME)=""  S (R,II)=0  ; % IS THE FIRST XREF  F  S II=$O(^DIC(4,II)) Q:II=""  Q:II="%"  D   . S X=^DIC(4,II,0),R=0   . S X1=$G(^DIC(4,II,99)) ;$G SETS TO NULL IF RECORD DOES NOT EXIST   . S F11=$P(X1,U,4) ;INACTIVE FLAG - POINTER   . Q:F11=1 ;QUIT IF F1=INACTIVE=1   . S F1=$G(^DIC(4,II,3))   . Q:F1=""   . S F2=$G(^DIC(4.1,F1,0))    . S F2N=$P(F2,U,1)   . S A1=$P(F2,U,1)    . S FACNAME=$P(X,U,1)   . I "VAMC"[F2N S R=1    . I "BVA/VBA-SO"[F2N S R=1   . I "RO"[F2N S R=1   . I "CENTRAL OFFICE"[F2N S R=1   . I "BVA"[F2N S R=1   . I "M&ROC"[F2N S R=1   . I "MC&RO"[F2N S R=1   . I "RO-OC"[F2N S R=1   . I "RO&IC"[F2N S R=1   . I "VAMROC"[F2N S R=1   . I "-RO"[F2N S R=1   . I "VAHSRO"[F2N S R=1   . I "TIGER TEAM"[F2N S R=1   . I II=460 S R=1   . Q:R=0   . S FACNAME=$P(X,U,1)   . S STATE=$P(X,U,2) I STATE="" Q:STATE=""   . S STATION=$P(X1,U,1)   . S FIEN=$P(X,U,1)   . S ST=$$EXTERNAL^DILFD(4,.02,"",STATE)   . S:SORTBY="STATE" RES(ST,II)=ST\_U\_FACNAME\_U\_STATION\_U\_II\_U\_A1   . S:SORTBY="NAME" RESULT(FACNAME)=ST\_U\_FACNAME\_U\_STATION\_U\_II  I SORTBY="STATE" S (R,II)=0,ST="" D   . N CNT S CNT=0   . F  S ST=$O(RES(ST)) Q:ST=""  D   .. S II=0   .. F  S II=$O(RES(ST,II)) Q:II=""  D   ... S CNT=$G(CNT)+1   ... S RESULT(CNT)=RES(ST,II)  Q** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBABEBD | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.1.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | [^DPT - [#2]](http://code.osehra.org/dox/Global_XkRQVA==.html)  ^DVB(396.25 –  [^DVB(396.3 - [#396.3]](http://code.osehra.org/dox/Global_XkRWQigzOTYuMw==.html)  [^TMP("DVBABEBD](http://code.osehra.org/dox/Global_XlRNUCgiRFZCQUJFQkQi.html) | $$FMTE^[*XLFDT*](http://code.osehra.org/dox/Routine_XLFDT.html)  DEM^[*VADPT*](http://code.osehra.org/dox/Routine_VADPT.html)  GETS^[*DIQ*](http://code.osehra.org/dox/Routine_DIQ.html) |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | GETS^[DIQ](http://code.osehra.org/dox/Routine_DIQ.html)  DEM^[VADPT](http://code.osehra.org/dox/Routine_VADPT.html)  $$FMTE^[XLFDT](http://code.osehra.org/dox/Routine_XLFDT.html) | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBABEBD ;ALB - CAPRI EXAM BY DATE REPORT;09/13/04 ; 2/10/11 12:12pm  ;;2.7;AMIE;\*\*57,108,185,190,192,193\*\*;Apr 10, 1995;Build 15  ;ALB/RTW - 192 added Claim Type,and Special Consideration(s)  Q  ; EXAMBYDT(MSG,BEGDT,ENDDT,DVBADLMTR) ; CALLED BY REMOTE PROCEDURE DVBAB EXAMS BY DATE  N DVBABCNT,DVBABIEN,DVBABPAT,DVBABT,DVBAB0,DVBABT1,DVBABT2,DVBABT3,DVBABT4,DVBABT5,DVBABT6,DFN  N DVBAA,DVBCNT,DVBX,DVBAD,DVBCTN,DVBCTW,DVBSC,DVBSCC,DVBSCN,DVBSCNS,DVBSCW,DVBSCWA  S DVBADLMTR=$S($G(DVBADLMTR)=1:",",1:0)  S DVBAD=DVBADLMTR  K ^TMP("DVBABEBD",DUZ) D SETSTAT  I $G(DVBAD)="," D  . S ^TMP("DVBABEBD",DUZ,1)="SSN"\_DVBAD\_"PATIENT NAME"\_DVBAD\_"REQUEST DATE"\_DVBAD\_"DATE RELEASED"\_DVBAD\_"DATE PRINTED BY RO"\_DVBAD\_"REQUEST STATUS"\_DVBAD\_"CLAIM TYPE"\_DVBAD\_"SPECIAL CONSIDERATION(S)"\_$C(13)  S DVBABCNT=$S(DVBAD'=",":1,1:2),DVBABIEN=0,MSG=$NA(^TMP("DVBABEBD",DUZ))  F  S DVBABIEN=$O(^DVB(396.3,DVBABIEN)) Q:+DVBABIEN=0 D  . ;REQUEST DATE  . S DVBAB0=$G(^DVB(396.3,DVBABIEN,0)) Q:DVBAB0=""  . S DVBABPAT=$G(^DPT($P(DVBAB0,"^"),0)) Q:DVBABPAT=""  . S DVBABT2=$P($P(DVBAB0,"^",2),".")  . I (DVBABT2>(BEGDT-1))&(DVBABT2<(ENDDT+1)) D  .. S DVBABT1=$P(DVBABPAT,"^",1) ;PATIENT NAME  .. S DFN=$P(DVBAB0,"^") D DEM^VADPT I $G(VADM(1))'="" S DVBABT6=$S($G(DVBAD)=",":$P($G(VADM(2)),U,2),1:$P($G(VADM(2)),U,1)) ;SSN  .. S DVBABT3=$P($P(DVBAB0,"^",14),".") ;DATE RELEASED  .. S DVBABT4=$P($P(DVBAB0,"^",16),".") ;DATE PRINTED BY RO  .. S DVBABT5=$P(DVBAB0,"^",18) ;REQUEST STATUS  .. S (DVBCTW,DVBSCWA)="" D CLAIMTYP,SPEC  .. S:$D(DVBABT(DVBABT5)) DVBABT5=DVBABT(DVBABT5)  .. S DVBABT2=$$FMTE^XLFDT(DVBABT2,"5DZ"),DVBABT3=$$FMTE^XLFDT(DVBABT3,"5DZ"),DVBABT4=$$FMTE^XLFDT(DVBABT4,"5DZ")  .. I $G(DVBAD)'="," S ^TMP("DVBABEBD",DUZ,DVBABCNT)=DVBABT1\_U\_DVBABT2\_U\_DVBABT3\_U\_DVBABT4\_U\_DVBABT5\_U\_DVBABT6\_U\_DVBCTW\_U\_DVBSCWA\_$C(13)  .. I $G(DVBAD)="," S ^TMP("DVBABEBD",DUZ,DVBABCNT)=DVBABT6\_DVBADLMTR\_""""\_DVBABT1\_""""\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT2,"5DZ")\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT3,"5DZ")\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT4,"5DZ") D  .. .S ^TMP("DVBABEBD",DUZ,DVBABCNT)=^TMP("DVBABEBD",DUZ,DVBABCNT)\_DVBADLMTR\_""""\_DVBABT5\_""""\_DVBAD\_""""\_DVBCTW\_""""\_DVBAD\_""""\_DVBSCWA\_""""\_$C(13)  .. S DVBABCNT=DVBABCNT+1  Q  ; SETSTAT S DVBABT("N")="NEW"  S DVBABT("P")="PENDING, REPORTED"  S DVBABT("S")="PENDING, SCHEDULED"  S DVBABT("R")="RELEASED TO RO, NOT PRINTED"  S DVBABT("C")="COMPLETED, PRINTED BY RO"  S DVBABT("X")="CANCELLED BY MAS"  S DVBABT("RX")="CANCELLED BY RO"  S DVBABT("T")="TRANSCRIBED"  S DVBABT("NT")="NEW, TRANSFERRED IN"  S DVBABT("CT")="COMPLETE, TRANSFERRED OUT"  Q CLAIMTYP ;THE CLAIM TYPE OF A 2507 REQUEST  S DVBCTW=""  Q:'$D(^DVB(396.3,DVBABIEN,9,0))  ;DVBIEN is the 2507 REQUEST FILE IEN  ;DVBCTW is the string /name of the CLAIM TYPE  D GETS^DIQ(396.3,DVBABIEN\_",","9.1\*","E","MSG","ERR")  S DVBCTW=MSG("396.32","1,"\_DVBABIEN\_",",".01","E")  Q  ; SPEC ;SPECIAL CONSIDERATION(S) FOR A 2507 REQUEST  K DVBSCW  S DVBSCWA=""  Q:'$D(^DVB(396.3,DVBABIEN,8))  ;DVBABIEN is the 2507 REQUEST FILE IEN  ;DVBSC is a the SPECIAL CONSIDERATION entry for the 2507 REQUEST  ;DVBSCN is the pointer number to the SPECIAL CONSIDERATION file 396.25  ;DVBSCW is the string /name of the SPECIAL CONSIDERATION  S DVBAA=$P(^DVB(396.3,DVBABIEN,8,0),U,4)  S (DVBSC,DVBCNT)=0 F  S DVBSC=$O(^DVB(396.3,DVBABIEN,8,DVBSC)) Q:'DVBSC  D  .S DVBSCN=$P(^DVB(396.3,DVBABIEN,8,DVBSC,0),U,1)  .S DVBSCW(DVBSC)=$G(^DVB(396.25,DVBSCN,0))  .S DVBCNT=DVBCNT+1  .I (DVBCNT'=DVBAA) S:$D(DVBSCW(DVBSC)) DVBSCW(DVBSC)=DVBSCW(DVBSC)\_","  S DVBX="" F  S DVBX=$O(DVBSCW(DVBX)) Q:'DVBX  S DVBSCWA=DVBSCWA\_DVBSCW(DVBX)  Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBABEBD ;ALB - CAPRI EXAM BY DATE REPORT;09/13/04 ; 2/10/11 12:12pm  ;;2.7;AMIE;\*\*57,108,185,190,192,**193**\*\*;Apr 10, 1995;Build 15  ;ALB/RTW - 192 added Claim Type,and Special Consideration(s)  Q  ; EXAMBYDT(MSG,BEGDT,ENDDT,DVBADLMTR) ; CALLED BY REMOTE PROCEDURE DVBAB EXAMS BY DATE  N DVBABCNT,DVBABIEN,DVBABPAT,DVBABT,DVBAB0,DVBABT1,DVBABT2,DVBABT3,DVBABT4,DVBABT5,DVBABT6,DFN  N DVBAA,DVBCNT,DVBX,DVBAD,DVBCTN,DVBCTW,DVBSC,DVBSCC,DVBSCN,DVBSCNS,DVBSCW,DVBSCWA  **; GRE; Reroute information added to report and CSV file**  **N RRIENINC,RRSTATINC,RRSTAT,RRDATE,RRTOSITE,RRFRMSITE,RRSTATDTE**  S DVBADLMTR=$S($G(DVBADLMTR)=1:",",1:0)  S DVBAD=DVBADLMTR **; AJF ; Request Status Conversion; removed call to SETSTAT**   K ^TMP("DVBABEBD",DUZ)  I $G(DVBAD)="," D  . S ^TMP("DVBABEBD",DUZ,1)="SSN"\_DVBAD\_"PATIENT NAME"\_DVBAD\_"REQUEST DATE"\_DVBAD\_"DATE RELEASED"\_DVBAD\_"DATE PRINTED BY RO"\_DVBAD\_"REQUEST STATUS"\_DVBAD\_"CLAIM TYPE"\_DVBAD\_"SPECIAL CONSIDERATION(S)"\_$C(13)  S DVBABCNT=$S(DVBAD'=",":1,1:2),DVBABIEN=0,MSG=$NA(^TMP("DVBABEBD",DUZ))  F  S DVBABIEN=$O(^DVB(396.3,DVBABIEN)) Q:+DVBABIEN=0 D  . ;REQUEST DATE  . S DVBAB0=$G(^DVB(396.3,DVBABIEN,0)) Q:DVBAB0=""  . S DVBABPAT=$G(^DPT($P(DVBAB0,"^"),0)) Q:DVBABPAT=""  . S DVBABT2=$P($P(DVBAB0,"^",2),".")  . I (DVBABT2>(BEGDT-1))&(DVBABT2<(ENDDT+1)) D  .. S DVBABT1=$P(DVBABPAT,"^",1) ;PATIENT NAME  .. S DFN=$P(DVBAB0,"^") D DEM^VADPT I $G(VADM(1))'="" S DVBABT6=$S($G(DVBAD)=",":$P($G(VADM(2)),U,2),1:$P($G(VADM(2)),U,1)) ;SSN  .. S DVBABT3=$P($P(DVBAB0,"^",14),".") ;DATE RELEASED  .. S DVBABT4=$P($P(DVBAB0,"^",16),".") ;DATE PRINTED BY RO  .. S DVBABT5=$P(DVBAB0,"^",18) ;REQUEST STATUS  .. S (DVBCTW,DVBSCWA)="" D CLAIMTYP,SPEC  **.. ;AJF ; Request Status Conversion  .. S DVBABT5=$$RTSTAT^DVBCUTL8(DVBABT5)**  .. S DVBABT2=$$FMTE^XLFDT(DVBABT2,"5DZ"),DVBABT3=$$FMTE^XLFDT(DVBABT3,"5DZ"),DVBABT4=$$FMTE^XLFDT(DVBABT4,"5DZ")  .. I $G(DVBAD)'="," S ^TMP("DVBABEBD",DUZ,DVBABCNT)=DVBABT1\_U\_DVBABT2\_U\_DVBABT3\_U\_DVBABT4\_U\_DVBABT5\_U\_DVBABT6\_U\_DVBCTW\_U\_DVBSCWA\_$C(13)  .. I $G(DVBAD)="," S ^TMP("DVBABEBD",DUZ,DVBABCNT)=DVBABT6\_DVBADLMTR\_""""\_DVBABT1\_""""\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT2,"5DZ")\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT3,"5DZ")\_DVBADLMTR\_$$FMTE^XLFDT(DVBABT4,"5DZ") D  .. .S ^TMP("DVBABEBD",DUZ,DVBABCNT)=^TMP("DVBABEBD",DUZ,DVBABCNT)\_DVBADLMTR\_""""\_DVBABT5\_""""\_DVBAD\_""""\_DVBCTW\_""""\_DVBAD\_""""\_DVBSCWA\_""""\_$C(13)  .. S DVBABCNT=DVBABCNT+1  Q  ; CLAIMTYP ;THE CLAIM TYPE OF A 2507 REQUEST  S DVBCTW=""  Q:'$D(^DVB(396.3,DVBABIEN,9,0))  ;DVBIEN is the 2507 REQUEST FILE IEN  ;DVBCTW is the string /name of the CLAIM TYPE  D GETS^DIQ(396.3,DVBABIEN\_",","9.1\*","E","MSG","ERR")  S DVBCTW=MSG("396.32","1,"\_DVBABIEN\_",",".01","E")  Q  ; SPEC ;SPECIAL CONSIDERATION(S) FOR A 2507 REQUEST  K DVBSCW  S DVBSCWA=""  Q:'$D(^DVB(396.3,DVBABIEN,8))  ;DVBABIEN is the 2507 REQUEST FILE IEN  ;DVBSC is a the SPECIAL CONSIDERATION entry for the 2507 REQUEST  ;DVBSCN is the pointer number to the SPECIAL CONSIDERATION file 396.25  ;DVBSCW is the string /name of the SPECIAL CONSIDERATION  S DVBAA=$P(^DVB(396.3,DVBABIEN,8,0),U,4)  S (DVBSC,DVBCNT)=0 F  S DVBSC=$O(^DVB(396.3,DVBABIEN,8,DVBSC)) Q:'DVBSC  D  .S DVBSCN=$P(^DVB(396.3,DVBABIEN,8,DVBSC,0),U,1)  .S DVBSCW(DVBSC)=$G(^DVB(396.25,DVBSCN,0))  .S DVBCNT=DVBCNT+1  .I (DVBCNT'=DVBAA) S:$D(DVBSCW(DVBSC)) DVBSCW(DVBSC)=DVBSCW(DVBSC)\_","  S DVBX="" F  S DVBX=$O(DVBSCW(DVBX)) Q:'DVBX  S DVBSCWA=DVBSCWA\_DVBSCW(DVBX)  Q  ; |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | DVBARSBD | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | DVBAB82 | CHK^[DIE](http://code.osehra.org/dox/Routine_DIE.html)  ($$GET1,GETS)  ^[DIQ](http://code.osehra.org/dox/Routine_DIQ.html)  ^[DIWP](http://code.osehra.org/dox/Routine_DIWP.html)  $$FMTE^[XLFDT](http://code.osehra.org/dox/Routine_XLFDT.html),  DEM^[VADPT](http://code.osehra.org/dox/Routine_VADPT.html) |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** |  | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** |  | | | | |
| **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 |
| --- |
| DVBARSBD ;ALB/RPM - CAPRI 2507 REQUEST STATUS BY DT RANGE REPORT ; 01/24/12  ;;2.7;AMIE;\*\*179,185,189,190,192\*\*;Apr 10, 1995;Build 15  ;  Q  ;NO DIRECT ENTRY  .  .  .  REQSTAT(DVBSDAT,DVBEDAT,DVBRSTAT,DVBDELIM,DVBNODT) ;entry for request status bye  ;  ; Input:  ; DVBSDAT - start date (FM format)  ; DVBEDAT - end date (FM format)  ; DVBRSTAT - request status (internal format)  ; DVBDELIM - return delimited results (0=no;1=yes)  ; DVBNODT - ignore date range (0=no;1=yes)  ;  N EXSTAT ;request status (external format)  N EXSDAT ;start date (external format: MM/DD/YYYY)  N EXEDAT ;end date (external format: MM/DD/YYYY)  N DVBARS ;request status conversion results  N DVBERR ;FM error msg  N DVBCNT ;returned record count  K ^TMP("DVBREQ",$J),^TMP("DVBREQN",$J)  S EXSDAT=$$FMTE^XLFDT(DVBSDAT,"5DZ")  S EXEDAT=$$FMTE^XLFDT(DVBEDAT,"5DZ")  S $P(DVBLINE,"-",131)=""  I DVBRSTAT="A" S EXSTAT="ALL"  E D  . D CHK^DIE(396.3,17,"E",DVBRSTAT,.DVBARS,"DVBERR")  . S EXSTAT=$G(DVBARS(0))  S DVBCNT=1  .  .  .   . F  S DVBIEN=$O(^DVB(396.3,"AH",CHGDAT,DVBIEN)) Q:'DVBIEN  D  . . S DVBSTAT=$$GET1^DIQ(396.3,DVBIEN\_",",17,"I","","")  . . I RSTAT="A"!(DVBSTAT=RSTAT) D  . . . K FLD  . . . I $$SETFLDS(DVBIEN,.FLD) D  . . . . S CNT=CNT+1,DVBXCNT=1  . . . . I $G(DVBAD)'="," S ^TMP("DVBREQ",$J,CNT)=FLD("IEN")\_U\_FLD("SS")\_U\_FLD("NM")\_U\_FLD("REQDT")\_U\_FLD("RELDT")\_U\_FLD("PRTDT")\_U\_FLD("RS")\_U\_FLD("CANDT")\_U\_FLD("RO")\_U\_FLD("CANRS",DVBXCNT)\_U\_FLD("CANCOM",DVBXCNT) D  . . . . . Q  .  .  .  SETFLDS(DVBIEN,DVBFLDS) ;build field array in external format  ;This function formats the collected record data in external format ; Integration Reference #10061 - DEM^VADPT  .  .  .  ;   N DFN  ;PATIENT file IEN used in VADPT call  N DVBDAT  ;2507 REQUEST data field array  N DVBIENS  ;FM IENS value  N DVBRSLT  ;function result  N VADM  ;VADPT return array  N DVBIEN4 ;the IEN FROM 2507 EXAM FILE 396.4  N DVBALAST ;number of lines in the wp cancellation comments  N DVBAI  ;for loop index  N DVBAX ;  .  .  .  . S DVBFLDS("IEN")=DVBIEN  . S DVBFLDS("NM")=$G(VADM(1))  . S DVBFLDS("SS")=$S(DVBDELIM:$P($G(VADM(2)),U,2),1:$P($G(VADM(2)),U,1))  . S DVBFLDS("RS")=$G(DVBDAT(396.3,DVBIENS,17,"E"))  . S DVBFLDS("REQDT")=$$FMTE^XLFDT($G(DVBDAT(396.3,DVBIENS,1,"I")),"5DZ")  . S DVBFLDS("RELDT")=$$FMTE^XLFDT($G(DVBDAT(396.3,DVBIENS,13,"I")),"5DZ")  . S DVBFLDS("PRTDT")=$$FMTE^XLFDT($G(DVBDAT(396.3,DVBIENS,15,"I")),"5DZ")  . S DVBFLDS("CANDT")=$$FMTE^XLFDT($G(DVBDAT(396.3,DVBIENS,19,"I")),"5DZ")  . S DVBFLDS("RO")=$G(DVBDAT(396.3,DVBIENS,2,"E"))  . D CLAIMTYP,SPEC  .  .  . DELIMHDR(EXSDAT,EXEDAT,EXSTAT) ;output delimited format header  ; Input:  ; EXSDAT - start date (external format)  ; EXEDAT - end date (external format)  ; EXSTAT - request status (external format)  .  .  .  Type"\_DVBAD\_"Special Consideration(s)",DVBCNT=DVBCNT+1  Q .  .  .  PLAINHDR(EXSDAT,EXEDAT,EXSTAT) ;output plain text header  .  .  .   U IO  S $P(DVBLINE,"-",131)=""  W "Date Range: "\_EXSDAT\_" - "\_EXEDAT  W !,"Request Status: ",EXSTAT  W !  W !,"SSN",?11,"PATIENT NAME",?33,"REQUEST DT",?45,"DT RELEASED"  W ?57,"DT PRINTED",?69,"STATUS",?98,"DT CANCELED",?110,"STATION"  W !,?2,"CLAIM TYPE"  W !,?2,"SPECIAL CONSIDERATION(S)"  W !,?2,"CANCELLATION REASON"  W !,?2,"CANCELLATION COMMENTS"  W !,DVBLINE  Q  ;  .  .  .  PLAIN ;output plain text format .  .  .  . . S DVBREQ=^TMP("DVBREQ",$J,DVBI,DVBIEN)  . . W !,$P(DVBREQ,U,2),?11,$E($P(DVBREQ,U,3),1,20),?33,$P(DVBREQ,U,4)  . . W ?45,$P(DVBREQ,U,5),?57,$P(DVBREQ,U,6),?69,$P(DVBREQ,U,7)  . . W ?98,$P(DVBREQ,U,8),?110,$E($P(DVBREQ,U,9),1,20),!  . . D CLAIMTYP,SPEC  . . W !,?2,"CLAIM TYPE: ",DVBCTW  . . W !,?2,"SPECIAL CONSIDERATION(S): ",DVBSCWA,!  . . S DVBIEN4=0 F  S DVBIEN4=$O(^DVB(396.4,"C",DVBIEN,DVBIEN4)) Q:'DVBIEN4  D  . . . I $D(^DVB(396.4,DVBIEN4,"CAN")) D  . . . . S DVBAY=($P($P(^DVB(396.4,DVBIEN4,"CAN"),"^",1),".",1)) I DVBAY>(DVBSDAT-1)&DVBAY<(DVBEDAT+1) D  . . . . . S DVBCNRS=$$GET1^DIQ(396.4,DVBIEN4,52) D  . . . . . .W !,?2,"CANCELLATION REASON: ",DVBCNRS,!  . . . . . I $D(^DVB(396.4,DVBIEN4,5)) D  . . . . . . K ^UTILITY($J,"W")  . . . . . . W !,?2,"CANCELLATION COMMENTS: " F LINE=0:0 S LINE=$O(^DVB(396.4,DVBIEN4,5,LINE)) Q:LINE=""  S X=^(LINE,0),DIWL=5,DIWR=75,DIWF="NW" D ^DIWP  . . . . . . W !  Q  .  .  .  GETRECSN(RSTAT,DVBCNT) ;collect 2507 REQUEST status matches and ignore date range  ;This procedure collects all 2507 REQUEST records that have a REQUEST STATUS .  .  .    . . . . S ^TMP("DVBREQ",$J,DVBCNT,DVBIEN)=^TMP("DVBREQ",$J,DVBCNT,DVBIEN)\_FLD("DVBCTW")\_U\_FLD("DVBSCWA")  . . . . S DVBCNT=DVBCNT+1,DVBXCNT=DVBXCNT+1  . . . I $G(DVBAD)="," D  . . . . S ^TMP("DVBREQ",$J,DVBCNT,DVBIEN)=FLD("SS")\_DVBAD\_""""\_FLD("NM")\_""""\_DVBAD\_FLD("REQDT")\_DVBAD\_FLD("RELDT")\_DVBAD\_""""\_FLD("PRTDT")\_""""\_DVBAD  . . . . S ^TMP("DVBREQ",$J,DVBCNT,DVBIEN)=^TMP("DVBREQ",$J,DVBCNT,DVBIEN)\_""""\_FLD("RS")\_""""\_DVBAD\_FLD("CANDT")\_DVBAD\_""""\_FLD("RO")\_""""\_DVBAD  . . . . S ^TMP("DVBREQ",$J,DVBCNT,DVBIEN)=^TMP("DVBREQ",$J,DVBCNT,DVBIEN)\_DVBAD\_DVBAD\_FLD("DVBCTW")\_DVBAD\_""""\_FLD("DVBSCWA")\_""""\_DVBAD  . . . . S DVBX=0 F  S DVBX=$O(FLD("IEN4",DVBX)) Q:'DVBX  D  . . . . . S ^TMP("DVBREQ",$J,DVBCNT,DVBIEN,DVBX)=FLD("SS")\_DVBAD\_DVBAD\_DVBAD\_DVBAD\_DVBAD\_DVBAD\_DVBAD\_DVBAD\_""""\_FLD("CANRS",DVBXCNT)\_""""\_DVBAD\_""""\_FLD("CANCOM",DVBXCNT)\_""""\_DVBAD\_$C(13)  . . . . . S DVBXCNT=DVBXCNT+1  . . . . S DVBCNT=DVBCNT+1  ;load output global with sorted list  S CHGDAT="",CNT=0 ;use "" because value could be zero ("0")  F  S CHGDAT=$O(^TMP("DVBREQN",$J,CHGDAT)) Q:(CHGDAT="") D  . S DVBIEN=0  . F  S DVBIEN=$O(^TMP("DVBREQN",$J,CHGDAT,DVBIEN)) Q:'DVBIEN  D  . . S CNT=CNT+1  . . S ^TMP("DVBREQ",$J,CNT)=^TMP("DVBREQN",$J,CHGDAT,DVBIEN)  Q |

| Modified Logic (Changes are in bold) |
| --- |
| DVBARSBD ;ALB/RPM - CAPRI 2507 REQUEST STATUS BY DT RANGE REPORT ; 01/24/12  ;;2.7;AMIE;\*\*179,185,189,190,192,**193**\*\*;Apr 10, 1995;Build 15  ;  Q  ;NO DIRECT ENTRY  **.**  **.**  **.**  N DVBERR ;FM error msg  N DVBCNT ;returned record count  **;GRE- 193 ;Request Status conversion Report changes ;**  **N RRIENINC ; re-route ien counter**  **N RRDATE ;re-route date**  **N RRTOSITE ;re-route to site**  **N RRFRMSITE ; re-route from site;**  **N RRSTATINC ; re-route status counter**  **N RRSTATDTE ; re-route status date**  **N RRSTAT ; re-route status**  **N DVBLINE ;header separator**  **;GRE ; end Request Status conversion Report changes ;**  K ^TMP("DVBREQ",$J),^TMP("DVBREQN",$J)  S EXSDAT=$$FMTE^XLFDT(DVBSDAT,"5DZ")  S EXEDAT=$$FMTE^XLFDT(DVBEDAT,"5DZ")  **S (RRDATE,RRFRMSITE,RRTOSITE,RRSTATDTE,RRSTAT)="",(RRIENINC,RRSTATINC)0**  S $P(DVBLINE,"-",131)=""  I DVBRSTAT="A" S EXSTAT="ALL"  E D  . D CHK^DIE(396.3,17,"E",DVBRSTAT,.DVBARS,"DVBERR")  . S EXSTAT=$G(DVBARS(0))  S DVBCNT=1  .  .  .   . F  S DVBIEN=$O(^DVB(396.3,"AH",CHGDAT,DVBIEN)) Q:'DVBIEN  D **. . ;AJF; Request Status Conversion**  . . S DVBSTAT=$$**RSTAT^DVBCUTL8($P(^DVB**(396.3,DVBIEN,0),U,18))  . . I RSTAT="A"!(DVBSTAT=RSTAT) D  . . . K FLD  . . . I $$SETFLDS(DVBIEN,.FLD) D  . . . . S CNT=CNT+1,DVBXCNT=1  . . . . I $G(DVBAD)'="," S ^TMP("DVBREQ**X**",$J,CNT)=FLD("IEN")\_U\_FLD("SS")\_U\_FLD("NM")\_U\_FLD("REQDT")\_U\_FLD("RELDT")\_U\_FLD("PRTDT")\_U\_FLD("RS")\_U\_FLD("CANDT")\_U\_FLD("RO")\_U\_FLD("CANRS",DVBXCNT)\_U\_FLD("CANCOM",DVBXCNT) D  . . . . .   . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)=FLD("IEN")\_U\_FLD("SS")\_U\_FLD("NM")\_U\_FLD("REQDT")\_U\_FLD("RELDT")\_U\_FLD("PRTDT")\_U\_FLD("RS")\_U\_FLD("CANDT")\_U\_FLD("RO")\_U\_FLD("CANRS",DVBXCNT)\_U\_FLD("CANCOM",DVBXCNT)\_U  . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)=^TMP("DVBREQ",$J,CNT,DVBIEN)\_FLD("DVBCTW")\_U\_FLD("DVBSCWA")**\_U\_RRDATE\_U\_RRFRMSITE\_U\_RRTOSITE**  . . . . . S CNT=CNT+1,DVBXCNT=*DVBXCNT*+1  . . . . I $G(DVBAD)="," D  . . . . . **S (RRCC,RRCR)=""**  . . . . . **I $D(FLD("IEN4",1)) S DVBX=0 F  S DVBX=$O(FLD("IEN4",DVBX)) Q:'DVBX  D**  . . . . . . **S RRCC=$S(RRCC="":FLD("CANRS",DVBX),1:RRCC\_"^"\_FLD("CANRS",DVBX))** **. . . . . . S RRCR=$S(RRCR="":FLD("CANCOM",DVBX),1:RRCR\_"^"\_FLD("CANCOM",DVBX))  . . . . . S FLD("CANRS")=RRCC,FLD("CANCOM")=RRCR**  . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)=FLD("SS")\_DVBAD\_""""\_FLD("NM")\_""""\_DVBAD\_FLD("REQDT")\_DVBAD\_FLD("RELDT")\_DVBAD\_""""\_FLD("PRTDT")\_""""\_DVBAD  . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)=^TMP("DVBREQ",$J,CNT,DVBIEN)\_""""\_FLD("RS")\_""""\_DVBAD\_FLD("CANDT")\_DVBAD\_""""\_FLD("RO")\_""""\_DVBAD  . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)=^TMP("DVBREQ",$J,CNT,DVBIEN**)\_""""\_FLD("CANRS")\_""""\_**DVBAD**\_FLD("CANCOM")\_**DVBAD**\_""""\_**FLD("DVBCTW")\_""""\_DVBAD  . . . . .  . . . . . S ^TMP("DVBREQ",$J,CNT,DVBIEN)**=^TMP("DVBREQ",$J,CNT,DVBIEN**)\_""""\_FLD("DVBSCWA")\_""""\_DVBAD\_FLD("RRDTE")\_DVBAD """"\_FLD("**RRSTE")\_""""\_DVBAD** **. . . . .** **S ^TMP("DVBREQ",$J,CNT,DVBIEN)=^TMP("DVBREQ",$J,CNT,DVBIEN)\_""""\_FLD("RRLOC")\_""""\_DVBAD**  **. . . . .** S CNT=CNT+1  Q  ;  SETFLDS(DVBIEN,DVBFLDS) ;build field array in external format  ;This function formats the collected record data in external format .  .  .  **;GE add re-route fields  ; DVBFLDS("RRDATE") - added with patch DVBA\*2.7\*193 holds the 2507 Re-Route Date  ; DVBFLDS("RRSITE") - added with patch DVBA\*2.7\*193 holds the 2507 Re-Route Site  ; DVBFLDS("RRLOC") - added with patch DVBA\*2.7\*193 holds the 2507 Re-Route Location .**  **.**  **.**  . D CLAIMTYP,SPEC**,GETRRDAT**. S DVBFLDS("DVBCTW")=DVBCTW  . S DVBFLDS("DVBSCWA")=DVBSCWA **. S DVBFLDS("RRDTE")=RRDATE  . S DVBFLDS("RRSTE")=RRFRMSITE  . S DVBFLDS("RRLOC")=RRTOSITE**  . S DVBXCNT=1  .  .  .  F DVBAI=2:1:DVBALAST S DVBFLDS("**CANRS**”,DVBXCNT)=DVBFLDS("CANCOM",DVBXCNT)\_WP(DVBAI,0)  . . . . . S DVBXCNT=DVBXCNT+1 .  .  .  DELIMHDR(EXSDAT,EXEDAT,EXSTAT) ;output delimited format header  ; Input:  ; EXSDAT - start date (external format)  ; EXEDAT - end date (external format)  ; EXSTAT - request status (external format)  **; GRE ; added re-route status information to the report**  **.**  **.**  **.** Type"\_DVBAD\_"Special Consideration(s**)"\_DVBAD\_"Re-Route Date"\_DVBAD\_"Re-Route Site"\_DVBAD\_"Re-Route Location"**,DVBCNT=DVBCNT+1  Q  .  .  .  PLAINHDR(EXSDAT,EXEDAT,EXSTAT) ;output plain text header .  .  .   U IO   W **!,"**Date Range: "\_EXSDAT\_" - "\_EXEDAT  W !,"Request Status: ",EXSTAT  W !,"-----------------------------------------------------------------------------------------------"  Q  . PLAIN ;output plain text format  .  .  .   . . S DVBREQ=^TMP("DVBREQ",$J,DVBI,DVBIEN) **. . D GETRRDAT ;GE 193  . . W !,DVBLINE ;GE 193  . . W !,"SSN:",?14,$P(DVBREQ,U,2) ;GE 193  . . W !,"PATIENT NAME:",?14,$E($P(DVBREQ,U,3),1,20) ;GE 193   . . W !,"REQUEST DT:",?14,$P(DVBREQ,U,4) ;GE 193  . . W !,"DT RELEASED:",?14,$P(DVBREQ,U,5) ;GE 193  . . W !,"DT PRINTED:",?14,$P(DVBREQ,U,6) ;GE 193  . . W !,"STATUS:",?14,$P(DVBREQ,U,7) ;GE 193  . . W !,"DT CANCELED:",?14,$P(DVBREQ,U,8) ;GE 193**  . . D CLAIMTYP,SPEC  . . W !,"CLAIM TYPE: ",DVBCTW  **. . W !,"SPECIAL CONSIDERATION(S):",DVBSCWA  . . W !,"RE-ROUTE DATE:",RRDATE       ; GE 193  . . W !,"RE-ROUTE SITE:",RRFRMSITE    ; GE 193  . . W !,"RE-ROUTE ROUTING LOCATION:",RRTOSITE  ; GE 193**  . . S DVBIEN4=0 F  S DVBIEN4=$O(^DVB(396.4,"C",DVBIEN,DVBIEN4)) Q:'DVBIEN4  D  . . . I $D(^DVB(396.4,DVBIEN4,"CAN")) D  . . . . S DVBAY=($P($P(^DVB(396.4,DVBIEN4,"CAN"),"^",1),".",1)) I DVBAY>(DVBSDAT-1)&DVBAY<(DVBEDAT+1) D  . . . . . S DVBCNRS=$$GET1^DIQ(396.4,DVBIEN4,52) D  . . . . . I DVBCNRS'="" D  **W !,"CANCELLATION REASON:",?14,DVBCNRS  ;GE 193**  . . . . . I $D(^DVB(396.4,DVBIEN4,5)) D  . . . . . . K ^UTILITY($J,"W")  . . . . . . W !,"CANCELLATION COMMENTS:",?14 F LINE=0:0 S LINE=$O(^DVB(396.4,DVBIEN4,5,LINE)) Q:LINE=""  S X=^(LINE,0),DIWL=5,DIWR=75,DIWF="NW" D ^DIWP  Q |

##### Templates – N/A for Requirement 2.6.1

##### Bulletins

Table 19: Bulletins

| Bulletins | Description | | | |
| --- | --- | --- | --- | --- |
| **Bulletin Name** | DVBA CAPRI REROUTE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | DVBAB1C |  |

| Routines | Description |
| --- | --- |
| **Mail Subject** | CAPRI: 2507 Exam Request Rejected |
| **Mail Group** | DVBA C 2507 REROUTE |
| **Parameters** |  |
| **Data Dictionary (DD) References** |  |

##### Unique Record(s) – N/A for Requirement 2.6.1

##### File or Global Size Changes – N/A for Requirement 2.6.1

##### Mail Groups

Table 24: Mail Groups

| Mail Groups | Activities | | | |
| --- | --- | --- | --- | --- |
| **Mail Group Name** | DVBA C 2507 REROUTE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | DVBAB1C |  |

| Mail Groups | Instructions | |
| --- | --- | --- |
| **Data Dictionary (DD) References** |  | |
| **Related Protocols** |  | |
| **Mail Group Description** | Will receive information when a 2507 Request is rejected. | |
| **Self-Enrollment Allowed** | Yes | No |
| **Type** | Public | Private |

##### Security Keys

Table 26: Security Keys

| Security Keys | Activities | | | |
| --- | --- | --- | --- | --- |
| **Security Key Name** | DVBA CAPRI REROUTEREASONS | | | |
| **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** | The holder of this is allowed to add/activate/inactivate the 2507 Reroute Code file | | | | |
| **Subordinate Keys** |  | | | | |
| **Mutually Exclusive Keys** |  | | | | |
| **Granting Condition Logic** |  | | | | |

| Current Logic |
| --- |
|  |

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

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

##### Options

Table28: Options

| Options | Activities | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Option Name** | DVBA CAPRI GUI | | | | | | | | | | |
| **Enhancement Category** | New | Modify | | | | Delete | | | No Change | | |
| **Associated Menu Options that will invoke this reference** | None | | | | | | | | | | |
| **Data Passing** | Input | | Output | | Both | | | Global Reference | | | Local Reference |
| **Menu Text Description** | This is the 'B' type option used by CAPRI GUI client application.  It contains all the RPCs used by the CAPRI GUI application. | | | | | | | | | | |
| **Option Type** | Edit | | | Print | | | Menu | | | Inquire | |
| Action | | | Run Routine | | | Other | | |  | |
| **Associated Routine** |  | | | | | | | | | | |
| **Option Definition** |  | | | | | | | | | | |
| Options | Activities | | | | | | | | | | |
| **Option Name** |  | | | | | | | | | | |
| **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** |  | | | | | | | | | | |
| **Option Definition** |  | | | | | | | | | | |

| 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 – N/A for Requirement 2.6.1

##### Remote Procedure Call (RPC)

Table 32: RPCs

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI GET DIV EXAM | | |
| **TAG^RTN** | CDIVE^DVBCUTL8 | | |
| **Input Parameters** | DIV | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC will return a the Default List of active exams. | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI GET DIVISION | | |
| **TAG^RTN** | CDIV^DVBCUTL8 | | |
| **Input Parameters** |  | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC will return all of the active division for this site. | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI GET REROUTE | | |
| **TAG^RTN** | REROST^DVBCUTL8 | | |
| **Input Parameters** | RSP | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC will return 1 if the Request is able to be rerouted or return 0  if the Request can not be rerouted. | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI GET REROUTE CODE | | |
| **TAG^RTN** | ARC^DVBCUTL8 | | |
| **Input Parameters** |  | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC returns the active Reroute Code from file #396.55 | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI REROUTE INFO | | |
| **TAG^RTN** | RINFO^DVBCUTL8 | | |
| **Input Parameters** | RIEN / RSTA | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC updates the status of the rerouted 2507 Request | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI REROUTE VAMC | | |
| **TAG^RTN** | GETFACNM^DVBACRVA | | |
| **Input Parameters** | SORTBY | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | VA FACALITIY NAMES AND STATES ARE RETURNED FOR ALL ENTERIES ON FILE. WILL BE  RE | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | DVBA CAPRI SEND REROUTE | | |
| **TAG^RTN** | EN^DVBCXFR1 | | |
| **Input Parameters** |  | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | This RPC reroute a 2507 C&P Request to another VAMC facility. | | |

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

##### Constants Defined in Interface – N/A for Requirement 2.6.1

##### Variables Defined in Interface – N/A for Requirement 2.6.1

##### Types Defined in Interface – N/A for Requirement 2.6.1

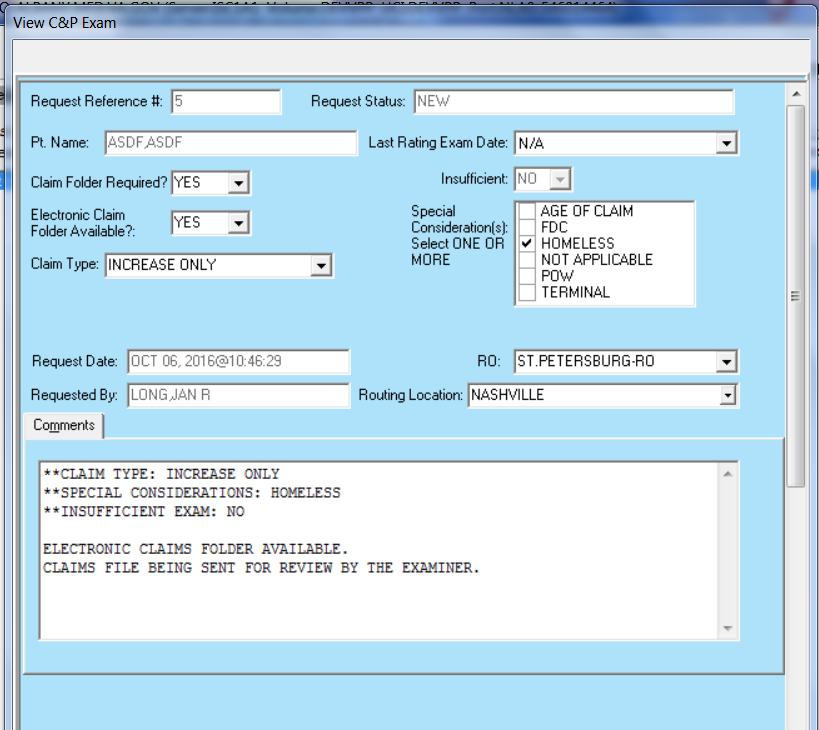
##### GUI – ViewExam.pas

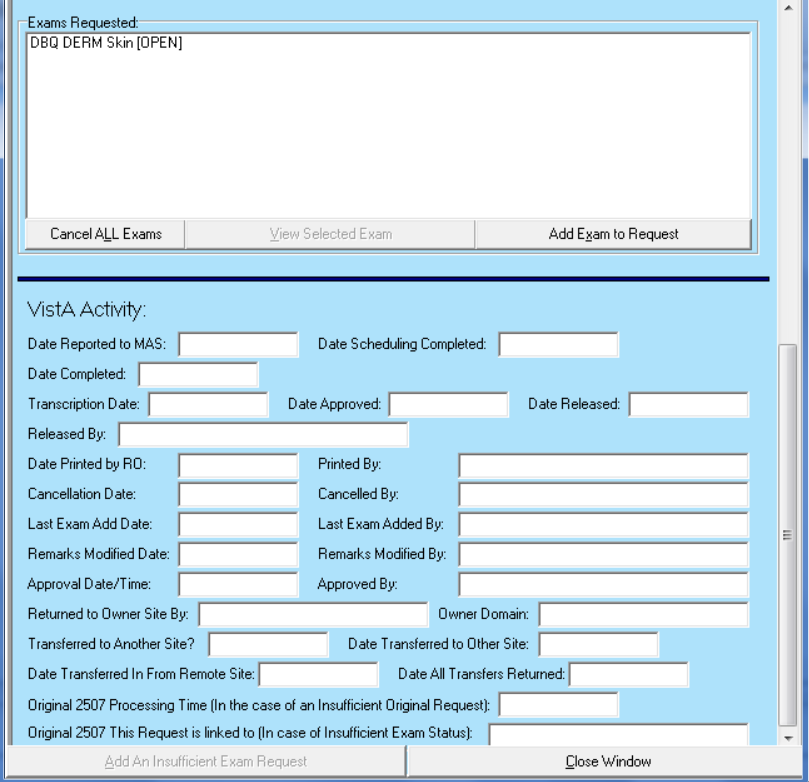
Table 36: GUI

| Unit Name | Description |
| --- | --- |
| ViewExam.pas | Modified ViewExam.pas to add re-routing functionality. |

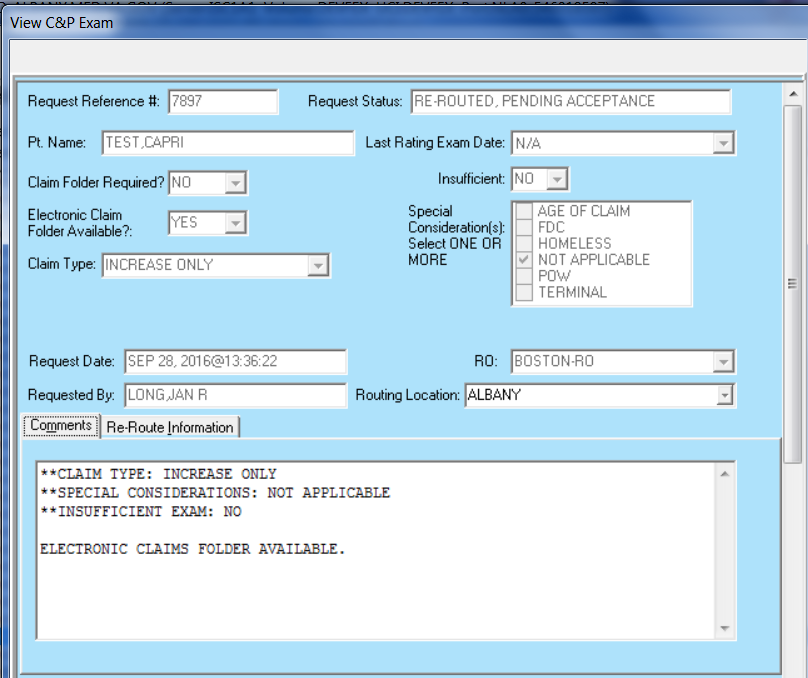
##### GUI Classes – N/A for Requirement 2.6.1

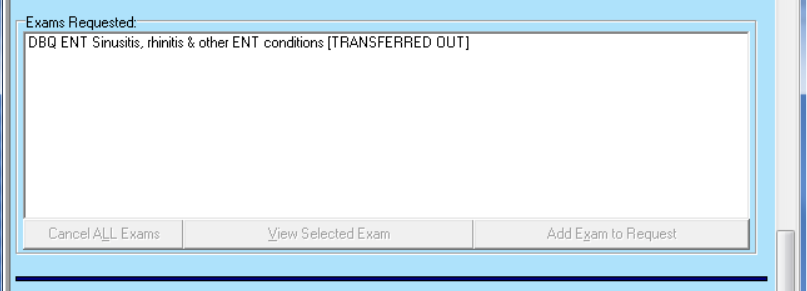
##### Current Form

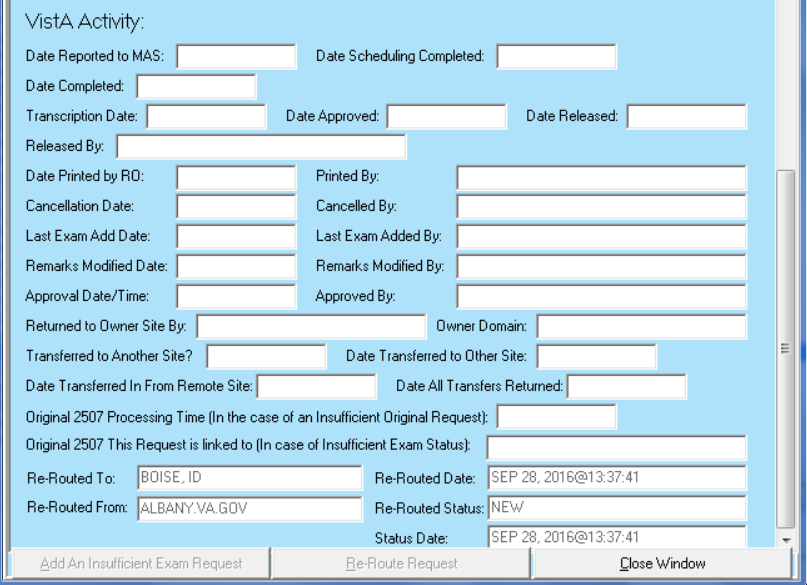




##### Modified Form







##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| pnlReRouteButtons: | TPanel; | Re-Route buttons panel |
| btnRejectReRoute: | TButton; | Reject Re-Route button |
| btnRerouteRequest: | TButton; | Re-Route Request button |
| pageCommentReRoute: | TPageControl; | Tab Control for Comments/Re-Route information |
| tsComments: | TTabSheet; | Comments Tab |
| tsReRouteInformation: | TTabSheet; | Re-Route Tab |
| btnAcceptReRoute: | TButton; | Accept Re-Route button |
| st508ReRouteReason: | TVA508StaticText; | Re-Route Reason label |
| st508ReRouteDescription: | TVA508StaticText; | Re-Route Description label |
| memoReRouteDescription: | TMemo; | Re-Route Description memo field |
| edtReRouteReason: | TEdit; | Re-Route Reason edit box |
| edtReRoutedStatusDate: | TEdit; | Re-Route Status Date edit box |
| edtReRoutedStatus: | TEdit; | Re-Route Status edit box |
| edtReRoutedTo: | TEdit; | Re-Routed To edit box |
| edtReRoutedDate: | TEdit; | Re-Routed Date edit box |
| edtReRoutedFrom: | TEdit; | Re-Routed From edit box |
| st508RejectReason: | TVA508StaticText; | Reject Reason label |
| memoRejectReason: | TMemo; | Reject Reason memo box |
| edtReRoutedToRoutingLoc: | TEdit; | Re-Routed To edit box |
| edtReRoutedFromRoutingLoc: | TEdit; | Re-Routed From edit box |
| st508ReroutedStatusDate: | TVA508StaticText; | Re-Routed Status Date label |
| st508ReroutedStatus: | TVA508StaticText; | Re-Routed Status label |
| st508ReRoutedDate: | TVA508StaticText; | Re-Routed Date label |
| st508ReRoutedTo: | TVA508StaticText; | Re-Routed To label |
| st508ReRoutedFrom: | TVA508StaticText; | Re-Routed From label |
| st508RoutingLocTo: | TVA508StaticText; | Routing Location To label |
| st508RoutingLocFrom: | TVA508StaticText; | Routing Location From label |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| edExamReferenceNumberChange | Edit field changed | Modified routine to see If Patch 193 is installed. If so, check to see if request has the correct statuses to be re-routed and enable or disable re-route request button accordingly. |
| btnRerouteRequestClick | Button clicked | Re-Route request button clicked. Compose request and call RPC to send. |
| FormActivate | Form Activate | Check for re-routed request and setup form accordingly. |
| btnAcceptReRouteClick | Button Clicked | Accept Re-Routed Request button clicked. |
| btnRejectReRouteClick | Button Clicked | Reject Re-Routed Request button clicked |
| FMRequestStatusChange | Edit box Changed | Request Status edit box changed. |
| btnInsufficientExamClick | Button Clicked | Insufficient Exam button clicked. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| CanRequestBeRerouted | Function | New. Call RPC to see if request can be re-routed. |
| UpdateCommentsReRouteTabs | Procedure | New. If the request is a newly re-routed request, we want to display the  Re-Route tab and Re-Route buttons. Highlight tab in orange to make it stand out. All controls should be read-only until the request is accepted/rejected. |
| UpdateControlsForEditingOrReadOnly | Procedure | New. Make controls read-only or editable based on re-route status. |
| IsThisANewReRoutedRequest | Function | New. Check for re-routed request and setup form accordingly. |
| UpdateReRoutedVistADisplayFields | Procedure | New. Retrieve and Populate Re-Route information in the VistA Activity section of View Exam form. |

##### Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
| Private ShowAcceptRejectReRouteButtons : Boolean | Private form variable |  |
| Public ReturnValue : String; | Public form variable |  |
| Public CurrentStationNumber : String | Public form variable |  |
| Public LocalReRoute : Boolean; | Public form variable |  |
| TAB\_REROUTE\_ACCEPT\_REJECT\_MODE = '1'; | Form Constant |  |
| TAB\_REROUTE\_DISPLAY\_INFO\_MODE = '2'; | Form Constant |  |
| CLOSE\_FORM = '1'; | Form Constant |  |
| DONT\_CLOSE\_FORM = '2'; | Form Constant |  |

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | ViewExam | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** | View C&P Requests | | | |

| Current Form Layout |
| --- |
| See current Form above |

| Modified Form Layout (Changes are in bold) |
| --- |
| See modified form above |

##### Functions

Table 48: Functions

| Function Name | FMRequestStatusChange | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified routine to look for New, Re-Routed status when looking for New status. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | edExamReferenceNumberChange | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmViewExam.FMRequestStatusChange(Sender: TObject);

begin

if ((FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') or (ReadOnlyMode = True)) then begin

(FMRequestStatus.Text <> 'PENDING, REPORTED') or

(ReadOnlyMode = True)) then begin

fmCommentsMemo.ReadOnly := True;

btnAddExam.Enabled := False;

fmOtherDisabilities1.Enabled := False;

fmOtherDisabilities2.Enabled := False;

fmOtherDisabilities3.Enabled := False;

fmRoutingLocation.Enabled := False;

fmRequestedBy.Enabled := False;

If IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

begin

cbClaimType.Enabled := False; // CodeCR666 JRL 11/5/14 Added Claim Type

chklstSpecialConsiderations.Enabled := False; // CodeCR667 JRL 11/5/14 Added Special Considerations

end

else // DVBA\*2.7\*187

fmPriorityofExam.Enabled := False;

fmOtherDocumentsRequired.Enabled := False;

fmClaimFolderRequired.Enabled := False;

fmElectronicClaimFolder.Enabled := False; // rpk 4/20/2015

FMRegionalOfficeNumber.Enabled := False; // rpk 4/20/2015

fmRoutingLocation.Enabled := False;

fmRegionalOfficeNumber.Enabled := False;

dtpLastRatingExamDate.Enabled := False; // -MER CodeCR116 8/2010

fmLastRatingExamDate.Enabled := False;

end

else

begin

fmCommentsMemo.ReadOnly := False;

btnAddExam.Enabled := True;

fmOtherDisabilities1.Enabled := True;

fmOtherDisabilities2.Enabled := True;

fmOtherDisabilities3.Enabled := True;

fmRoutingLocation.Enabled := True;

fmRequestedBy.Enabled := False; // Changed this

fmOtherDocumentsRequired.Enabled := True;

fmClaimFolderRequired.Enabled := True;

fmElectronicClaimFolder.Enabled := True; // rpk 4/20/2015

If IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

begin

cbClaimType.Enabled := True; // CodeCR666 JRL 1/29/15 Added Claim Type

chklstSpecialConsiderations.Enabled := True; // CodeCR667 JRL 11/5/14 Added Special Considerations

end

else // DVBA\*2.7\*187

fmPriorityofExam.Enabled := True; // CodeCR666 JRL 11/5/14 Removed Priority of Exam

// fmRegionalOfficeNumber.Enabled := False;

FMRegionalOfficeNumber.Enabled := True; // rpk 4/20/2015

dtpLastRatingExamDate.Enabled := True; // -MER CodeCR116 8/2010

fmLastRatingExamDate.Enabled := True;

end;

Set508ControlOptions; // -MER CodeCR116 8/2010

//SaveFlag:=True;

end;

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

procedure TfrmViewExam.FMRequestStatusChange(Sender: TObject);

begin

**// if ((FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') or (ReadOnlyMode = True)) then begin**

**if ((Copy(FMRequestStatus.Text,1,3) <> 'NEW') and // Patch 193, treat NEW, RE-ROUTED status the same as NEW**

(FMRequestStatus.Text <> 'PENDING, REPORTED') or

(ReadOnlyMode = True)) then begin

fmCommentsMemo.ReadOnly := True;

btnAddExam.Enabled := False;

fmOtherDisabilities1.Enabled := False;

fmOtherDisabilities2.Enabled := False;

fmOtherDisabilities3.Enabled := False;

fmRoutingLocation.Enabled := False;

fmRequestedBy.Enabled := False;

If IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

begin

cbClaimType.Enabled := False; // CodeCR666 JRL 11/5/14 Added Claim Type

chklstSpecialConsiderations.Enabled := False; // CodeCR667 JRL 11/5/14 Added Special Considerations

end

else // DVBA\*2.7\*187

fmPriorityofExam.Enabled := False;

fmOtherDocumentsRequired.Enabled := False;

fmClaimFolderRequired.Enabled := False;

fmElectronicClaimFolder.Enabled := False; // rpk 4/20/2015

FMRegionalOfficeNumber.Enabled := False; // rpk 4/20/2015

fmRoutingLocation.Enabled := False;

fmRegionalOfficeNumber.Enabled := False;

dtpLastRatingExamDate.Enabled := False; // -MER CodeCR116 8/2010

fmLastRatingExamDate.Enabled := False;

end

else

begin

fmCommentsMemo.ReadOnly := False;

btnAddExam.Enabled := True;

fmOtherDisabilities1.Enabled := True;

fmOtherDisabilities2.Enabled := True;

fmOtherDisabilities3.Enabled := True;

fmRoutingLocation.Enabled := True;

fmRequestedBy.Enabled := False; // Changed this

fmOtherDocumentsRequired.Enabled := True;

fmClaimFolderRequired.Enabled := True;

fmElectronicClaimFolder.Enabled := True; // rpk 4/20/2015

If IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

begin

cbClaimType.Enabled := True; // CodeCR666 JRL 1/29/15 Added Claim Type

chklstSpecialConsiderations.Enabled := True; // CodeCR667 JRL 11/5/14 Added Special Considerations

end

else // DVBA\*2.7\*187

fmPriorityofExam.Enabled := True; // CodeCR666 JRL 11/5/14 Removed Priority of Exam

// fmRegionalOfficeNumber.Enabled := False;

FMRegionalOfficeNumber.Enabled := True; // rpk 4/20/2015

dtpLastRatingExamDate.Enabled := True; // -MER CodeCR116 8/2010

fmLastRatingExamDate.Enabled := True;

end;

Set508ControlOptions; // -MER CodeCR116 8/2010

//SaveFlag:=True;

end;

Table 48: Functions

| Function Name | btnInsufficientExamClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified routine to update Request Status field to new field type. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnInsufficientExamClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmViewExam.btnInsufficientExamClick(Sender: TObject);

var

x, y: integer;

IFlag: Boolean;

tempExam: string;

begin

IFlag := False;

if lstExamsRequested.Items.Count > 0 then begin

for x := 0 to lstExamsRequested.Items.Count - 1 do

if Length(lstExamsRequested.Items[x]) > Pos('] ', lstExamsRequested.Items[x]) + 3 then

IFlag := True;

if IFLag = False then begin

ShowMessageCAPRI('You must mark at least one exam insufficient to perform this action.');

exit;

end;

end;

InsufficientExam := True;

AddRequestIEN := '+1';

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := 'DVBAB CHECK CREDENTIALS';

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do

ShowMessageCAPRI('RPC DVBAB CHECK CREDENTIALS could not be accessed!');

end;

if frmMain.RPCBroker1.Results[0] <> '[OK]' then begin

ShowMessageCAPRI(frmMain.RPCBroker1.Results[0]);

application.terminate;

Exit;

end;

frmMain.btnAddRequest.Enabled := False;

frmMain.btnViewExam.Enabled := False;

// If credentials are ok, then continue

// Populate lists and controls

frmNewExam := TfrmNewExam.Create(frmMain);

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

frmNewExam.Caption := FORM\_TITLE\_INSUFFICIENT\_EXAM; //CodeCR669 JRL 10/7/14

frmNewExam.FMInsufficientExam.ItemIndex := 1; // 0 = No, 1 = Yes //CodeCR669 JRL 10/7/14

frmNewExam.FMInsufficientExam.Enabled := FALSE; //CodeCR669 JRL 10/7/14

end;

frmMain.FMListerExams.GetList(frmNewExam.FMExamsList.Items);

frmNewExam.FMExamsList.GetList;

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

if frmNewExam.FMExamsList.Items.Count > 0 then

for x := 0 to frmNewExam.FMExamsList.Items.Count - 1 do

if Pos(' I', frmNewExam.FMExamsList.Items[x]) = 0 then

frmNewExam.CheckBoxExamsToComplete.Items.Add(Copy(frmNewExam.FMExamsList.Items[x], 1, Pos(' A', frmNewExam.FMExamsList.Items[x]) - 3));

frmMain.FMListerMedicalCenterDivision.GetList(frmNewExam.FMRoutingLocation.Items);

frmNewExam.FMRoutingLocation.Items := listMedicalCenterDivision;

//frmMain.FMListerInstitutionFile.GetList(frmNewExam.FMRegionalOfficeNumber.Items);

frmNewExam.FMRegionalOfficeNumber.Items.Clear;

if frmMain.lstInstitutions.Items.Count > 0 then

for x := 0 to frmMain.lstInstitutions.Items.Count - 1 do

frmNewExam.FMRegionalOfficeNumber.Items.Add(Copy(frmMain.lstInstitutions.Items[x], 1, Pos('^', frmMain.lstInstitutions.Items[x]) - 1));

frmNewExam.FMPatientName.Text := Copy(frmMain.Panel1.caption, 1, Pos(' SSN#', frmMain.Panel1.Caption));

frmNewExam.FMPatientNameIEN.Text := '`' + PatientIEN;

frmNewExam.FMRequestDate.Text := 'NOW';

frmNewExam.FMRequestedBy.Text := frmMain.FMUserName.Caption;

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

frmNewExam.FMRequestStatus.Text := 'N';

frmNewExam.FMClaimFolderRequired.Text := '';

frmNewExam.FMElectronicClaimFolder.Text := ''; // CodeCR704 JRL 4/17/15

frmNewExam.FMOtherDocumentsRequired.Text := '';

frmNewExam.FMRegionalOfficeNumberIEN.Text := FMRegionalOfficeNumberIEN.Text;

frmNewExam.FMRegionalOfficeNumber.Text := FMRegionalOfficeNumber.Text;

frmNewExam.FMRegionalOfficeNumber.Enabled := False;

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

frmNewExam.FMPriorityOfExam.Text := '';

frmNewExam.FMOtherDisabilities1.Text := '';

frmNewExam.FMOtherDisabilities2.Text := '';

frmNewExam.FMOtherDisabilities3.Text := '';

frmNewExam.FMRoutingLocation.Text := '';

frmNewExam.FMLastRatingExamDate.Text := '';

frmNewExam.FMCommentsMemo.Lines.Clear;

frmNewExam.FMOriginal2507Request.Text := '`' + EdExamReferenceNumber.text;

// frmNewExam.FMOriginal2507ProcessingTime.Text := FMEdit10.Text;

frmNewExam.FMOriginal2507ProcessingTime.Text := FMEditElapsedTime.Text; // rpk 4/20/2015

if IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

begin

LoadClaimTypeControlValues; // CodeCR666 JRL 11/12/14

LoadSpecialConsiderationControlValues; // CodeCR667 JRL 11/12/14

end

else // DVBA\*2.7\*187

begin

frmNewExam.FMPriorityOfExam.Enabled := False; // Don't less user change the status

frmNewExam.FMPriorityOfExam.Text := 'INSUFFICIENT EXAM';

frmNewExam.FMPriorityOfExamExit(Application);

end;

//Setup for new entry

frmNewExam.FMPatientNameIEN.IENS := '+1';

frmNewExam.FMRequestDate.IENS := '+1';

frmNewExam.FMClaimFolderRequiredFake.IENS := '+1';

if IsPatchInstalled('DVBA\*2.7\*189') then

frmNewExam.FMInsufficientExamFake.IENS := '+1'; // CodeCR669 JRL 11/19/14

frmNewExam.FMOtherDocumentsRequiredFake.IENS := '+1';

frmNewExam.FMRegionalOfficeNumberIEN.IENS := '+1';

frmNewExam.FMRequestedByIEN.IENS := '+1';

frmNewExam.FMRequestStatus.IENS := '+1';

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

frmNewExam.FMPriorityOfExamFake.IENS := '+1';

frmNewExam.FMOtherDisabilities1.IENS := '+1';

frmNewExam.FMOtherDisabilities2.IENS := '+1';

frmNewExam.FMOtherDisabilities3.IENS := '+1';

frmNewExam.FMRoutingLocationFake.IENS := '+1';

frmNewExam.FMLastRatingExamDate.IENS := '+1';

frmNewExam.FMCommentsMemo.IENS := '+1';

frmNewExam.FMOriginal2507Request.IENS := '+1';

frmNewExam.FMOriginal2507ProcessingTime.IENS := '+1';

//Enter Request

frmNewExam.Height := frmMain.Height - 75;

if frmNewExam.Height > 698 then

frmNewExam.Height := 698;

frmNewExam.Top := 7 + frmMain.Top + ((frmMain.Height - frmNewExam.Height) div 2);

frmNewExam.Left := frmMain.Left + ((frmMain.Width - frmNewExam.Width) div 2);

frmNewExam.btnSendRequest.Top := frmNewExam.Height - 73;

frmNewExam.btnCancel.Top := frmNewExam.Height - 73;

frmNewExam.Scrollbox1.Height := frmNewExam.Height - 74;

frmNewExam.ListBoxInsufficientExams.Items.Clear;

//Enter request

//frmNewExam.Top:=frmMain.Top + ((frmMain.Height-frmNewExam.Height) div 2);

//frmNewExam.Left:=frmMain.Left + ((frmMain.Width-frmNewExam.Width) div 2);

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

if frmNewExam.FMExamsList.Items.Count > 0 then

for x := 0 to frmViewExam.FMExamsList.Items.Count - 1 do

if Pos(' I', frmViewExam.FMExamsList.Items[x]) = 0 then

frmNewExam.CheckBoxExamsToComplete.Items.Add(Copy(frmViewExam.FMExamsList.Items[x], 1, Pos(' A', frmViewExam.FMExamsList.Items[x]) - 3));

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do begin

TempExam := Copy(lstExamsRequested.Items[x], 1, pos(' [C', lstExamsRequested.Items[x]) - 1);

for y := frmNewExam.CheckBoxExamsToComplete.Items.Count - 1 downto 0 do

if frmNewExam.CheckBoxExamsToComplete.Items[y] = TempExam then

frmNewExam.CheckBoxExamsToComplete.Checked[y] := true;

end;

for y := frmNewExam.CheckBoxExamsToComplete.Items.Count - 1 downto 0 do

if frmNewExam.CheckBoxExamsToComplete.Checked[y] = True then begin

frmNewExam.CheckBoxExamsToComplete.Checked[y] := False;

frmNewExam.ListBoxInsufficientExams.Items.Add(frmNewExam.CheckBoxExamsToComplete.Items[y]);

end

else begin

frmNewExam.CheckBoxExamsToComplete.Items.Delete(y);

end;

frmNewExam.CheckboxExamsToComplete.Items.Clear;

frmNewExam.Scrollbox1.VertScrollBar.Position := 0;

if frmNewExam.FMRegionalOfficeNumberIEN.Text <> '' then

frmNewExam.FMRoutingLocation.Enabled := True;

frmNewExam.FMListerCustomExamListDivisionsListBox.GetList;

//Check for inactive divisions

if frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count > 0 then

for x := 0 to frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count - 1 do begin

frmNewExam.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

frmNewExam.CustomExamListDivisions.IENS := frmNewExam.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmNewExam.CustomExamListDivisions.GetAndFill;

//Inactivate if "YES"

if frmNewExam.FMEditInactivateDivision.Text = 'YES' then begin

if frmNewExam.FMRoutingLocation.Items.Count > 0 then

for y := frmNewExam.FMRoutingLocation.Items.Count - 1 downto 0 do begin

frmNewExam.FMRoutingLocation.ItemIndex := y;

if Piece(frmNewExam.FMRoutingLocation.Items[y], '^', 2) = frmNewExam.FMListerCustomExamListDivisionsListBox.Items[x] then begin

frmNewExam.FMRoutingLocation.Items.Delete(y);

end;

end;

end;

end;

frmNewExam.FMRoutingLocation.ItemIndex := -1;

FrmNewExam.ShowModal;

frmNewExam.Release;

frmNewExam := nil;

frmMain.ExamRequestRefreshClick(Application);

frmMain.btnAddRequest.Enabled := True;

if frmMain.FMExamRequestListbox.ItemIndex > -1 then

frmMain.btnViewExam.Enabled := True;

InsufficientExam := False; // Reset Flag

lstExamsRequested.ItemIndex := -1; // Reset List

end;

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

procedure TfrmViewExam.btnInsufficientExamClick(Sender: TObject);

var

x, y: integer;

IFlag: Boolean;

tempExam: string;

begin

IFlag := False;

if lstExamsRequested.Items.Count > 0 then begin

for x := 0 to lstExamsRequested.Items.Count - 1 do

if Length(lstExamsRequested.Items[x]) > Pos('] ', lstExamsRequested.Items[x]) + 3 then

IFlag := True;

if IFLag = False then begin

ShowMessageCAPRI('You must mark at least one exam insufficient to perform this action.');

exit;

end;

end;

InsufficientExam := True;

AddRequestIEN := '+1';

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := 'DVBAB CHECK CREDENTIALS';

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do

ShowMessageCAPRI('RPC DVBAB CHECK CREDENTIALS could not be accessed!');

end;

if frmMain.RPCBroker1.Results[0] <> '[OK]' then begin

ShowMessageCAPRI(frmMain.RPCBroker1.Results[0]);

application.terminate;

Exit;

end;

frmMain.btnAddRequest.Enabled := False;

frmMain.btnViewExam.Enabled := False;

// If credentials are ok, then continue

// Populate lists and controls

frmNewExam := TfrmNewExam.Create(frmMain);

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

frmNewExam.Caption := FORM\_TITLE\_INSUFFICIENT\_EXAM; //CodeCR669 JRL 10/7/14

frmNewExam.FMInsufficientExam.ItemIndex := 1; // 0 = No, 1 = Yes //CodeCR669 JRL 10/7/14

frmNewExam.FMInsufficientExam.Enabled := FALSE; //CodeCR669 JRL 10/7/14

end;

frmMain.FMListerExams.GetList(frmNewExam.FMExamsList.Items);

frmNewExam.FMExamsList.GetList;

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

if frmNewExam.FMExamsList.Items.Count > 0 then

for x := 0 to frmNewExam.FMExamsList.Items.Count - 1 do

if Pos(' I', frmNewExam.FMExamsList.Items[x]) = 0 then

frmNewExam.CheckBoxExamsToComplete.Items.Add(Copy(frmNewExam.FMExamsList.Items[x], 1, Pos(' A', frmNewExam.FMExamsList.Items[x]) - 3));

frmMain.FMListerMedicalCenterDivision.GetList(frmNewExam.FMRoutingLocation.Items);

frmNewExam.FMRoutingLocation.Items := listMedicalCenterDivision;

//frmMain.FMListerInstitutionFile.GetList(frmNewExam.FMRegionalOfficeNumber.Items);

frmNewExam.FMRegionalOfficeNumber.Items.Clear;

if frmMain.lstInstitutions.Items.Count > 0 then

for x := 0 to frmMain.lstInstitutions.Items.Count - 1 do

frmNewExam.FMRegionalOfficeNumber.Items.Add(Copy(frmMain.lstInstitutions.Items[x], 1, Pos('^', frmMain.lstInstitutions.Items[x]) - 1));

frmNewExam.FMPatientName.Text := Copy(frmMain.Panel1.caption, 1, Pos(' SSN#', frmMain.Panel1.Caption));

frmNewExam.FMPatientNameIEN.Text := '`' + PatientIEN;

frmNewExam.FMRequestDate.Text := 'NOW';

frmNewExam.FMRequestedBy.Text := frmMain.FMUserName.Caption;

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

**frmNewExam.FMRequestStatus.Text := 'NEW' // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

**else // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

frmNewExam.FMRequestStatus.Text := 'N'; // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE

frmNewExam.FMClaimFolderRequired.Text := '';

frmNewExam.FMElectronicClaimFolder.Text := ''; // CodeCR704 JRL 4/17/15

frmNewExam.FMOtherDocumentsRequired.Text := '';

frmNewExam.FMRegionalOfficeNumberIEN.Text := FMRegionalOfficeNumberIEN.Text;

frmNewExam.FMRegionalOfficeNumber.Text := FMRegionalOfficeNumber.Text;

frmNewExam.FMRegionalOfficeNumber.Enabled := False;

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

frmNewExam.FMPriorityOfExam.Text := '';

frmNewExam.FMOtherDisabilities1.Text := '';

frmNewExam.FMOtherDisabilities2.Text := '';

frmNewExam.FMOtherDisabilities3.Text := '';

frmNewExam.FMRoutingLocation.Text := '';

frmNewExam.FMLastRatingExamDate.Text := '';

frmNewExam.FMCommentsMemo.Lines.Clear;

frmNewExam.FMOriginal2507Request.Text := '`' + EdExamReferenceNumber.text;

// frmNewExam.FMOriginal2507ProcessingTime.Text := FMEdit10.Text;

frmNewExam.FMOriginal2507ProcessingTime.Text := FMEditElapsedTime.Text; // rpk 4/20/2015

if IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

begin

LoadClaimTypeControlValues; // CodeCR666 JRL 11/12/14

LoadSpecialConsiderationControlValues; // CodeCR667 JRL 11/12/14

end

else // DVBA\*2.7\*187

begin

frmNewExam.FMPriorityOfExam.Enabled := False; // Don't less user change the status

frmNewExam.FMPriorityOfExam.Text := 'INSUFFICIENT EXAM';

frmNewExam.FMPriorityOfExamExit(Application);

end;

//Setup for new entry

frmNewExam.FMPatientNameIEN.IENS := '+1';

frmNewExam.FMRequestDate.IENS := '+1';

frmNewExam.FMClaimFolderRequiredFake.IENS := '+1';

if IsPatchInstalled('DVBA\*2.7\*189') then

frmNewExam.FMInsufficientExamFake.IENS := '+1'; // CodeCR669 JRL 11/19/14

frmNewExam.FMOtherDocumentsRequiredFake.IENS := '+1';

frmNewExam.FMRegionalOfficeNumberIEN.IENS := '+1';

frmNewExam.FMRequestedByIEN.IENS := '+1';

frmNewExam.FMRequestStatus.IENS := '+1';

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/5/15 - Added test for backward compatibility

frmNewExam.FMPriorityOfExamFake.IENS := '+1';

frmNewExam.FMOtherDisabilities1.IENS := '+1';

frmNewExam.FMOtherDisabilities2.IENS := '+1';

frmNewExam.FMOtherDisabilities3.IENS := '+1';

frmNewExam.FMRoutingLocationFake.IENS := '+1';

frmNewExam.FMLastRatingExamDate.IENS := '+1';

frmNewExam.FMCommentsMemo.IENS := '+1';

frmNewExam.FMOriginal2507Request.IENS := '+1';

frmNewExam.FMOriginal2507ProcessingTime.IENS := '+1';

//Enter Request

frmNewExam.Height := frmMain.Height - 75;

if frmNewExam.Height > 698 then

frmNewExam.Height := 698;

frmNewExam.Top := 7 + frmMain.Top + ((frmMain.Height - frmNewExam.Height) div 2);

frmNewExam.Left := frmMain.Left + ((frmMain.Width - frmNewExam.Width) div 2);

frmNewExam.btnSendRequest.Top := frmNewExam.Height - 73;

frmNewExam.btnCancel.Top := frmNewExam.Height - 73;

frmNewExam.Scrollbox1.Height := frmNewExam.Height - 74;

frmNewExam.ListBoxInsufficientExams.Items.Clear;

//Enter request

//frmNewExam.Top:=frmMain.Top + ((frmMain.Height-frmNewExam.Height) div 2);

//frmNewExam.Left:=frmMain.Left + ((frmMain.Width-frmNewExam.Width) div 2);

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

if frmNewExam.FMExamsList.Items.Count > 0 then

for x := 0 to frmViewExam.FMExamsList.Items.Count - 1 do

if Pos(' I', frmViewExam.FMExamsList.Items[x]) = 0 then

frmNewExam.CheckBoxExamsToComplete.Items.Add(Copy(frmViewExam.FMExamsList.Items[x], 1, Pos(' A', frmViewExam.FMExamsList.Items[x]) - 3));

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do begin

TempExam := Copy(lstExamsRequested.Items[x], 1, pos(' [C', lstExamsRequested.Items[x]) - 1);

for y := frmNewExam.CheckBoxExamsToComplete.Items.Count - 1 downto 0 do

if frmNewExam.CheckBoxExamsToComplete.Items[y] = TempExam then

frmNewExam.CheckBoxExamsToComplete.Checked[y] := true;

end;

for y := frmNewExam.CheckBoxExamsToComplete.Items.Count - 1 downto 0 do

if frmNewExam.CheckBoxExamsToComplete.Checked[y] = True then begin

frmNewExam.CheckBoxExamsToComplete.Checked[y] := False;

frmNewExam.ListBoxInsufficientExams.Items.Add(frmNewExam.CheckBoxExamsToComplete.Items[y]);

end

else begin

frmNewExam.CheckBoxExamsToComplete.Items.Delete(y);

end;

frmNewExam.CheckboxExamsToComplete.Items.Clear;

frmNewExam.Scrollbox1.VertScrollBar.Position := 0;

if frmNewExam.FMRegionalOfficeNumberIEN.Text <> '' then

frmNewExam.FMRoutingLocation.Enabled := True;

frmNewExam.FMListerCustomExamListDivisionsListBox.GetList;

//Check for inactive divisions

if frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count > 0 then

for x := 0 to frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count - 1 do begin

frmNewExam.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

frmNewExam.CustomExamListDivisions.IENS := frmNewExam.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmNewExam.CustomExamListDivisions.GetAndFill;

//Inactivate if "YES"

if frmNewExam.FMEditInactivateDivision.Text = 'YES' then begin

if frmNewExam.FMRoutingLocation.Items.Count > 0 then

for y := frmNewExam.FMRoutingLocation.Items.Count - 1 downto 0 do begin

frmNewExam.FMRoutingLocation.ItemIndex := y;

if Piece(frmNewExam.FMRoutingLocation.Items[y], '^', 2) = frmNewExam.FMListerCustomExamListDivisionsListBox.Items[x] then begin

frmNewExam.FMRoutingLocation.Items.Delete(y);

end;

end;

end;

end;

frmNewExam.FMRoutingLocation.ItemIndex := -1;

FrmNewExam.ShowModal;

frmNewExam.Release;

frmNewExam := nil;

frmMain.ExamRequestRefreshClick(Application);

frmMain.btnAddRequest.Enabled := True;

if frmMain.FMExamRequestListbox.ItemIndex > -1 then

frmMain.btnViewExam.Enabled := True;

InsufficientExam := False; // Reset Flag

lstExamsRequested.ItemIndex := -1; // Reset List

end;

Table 48: Functions

| Function Name | edExamReferenceNumberChange | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified routine to see If Patch 193 is installed. If so, check to see if request has the correct statuses to be re-routed and enable or disable re-route request button accordingly. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | edExamReferenceNumberChange | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmViewExam.edExamReferenceNumberChange(Sender: TObject);

begin

Scrollbox1.VertScrollbar.Position := 0;

SaveFlag := False; // Set flags to false

CommentsFlag := False;

if (FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') then

if not IsPatchInstalled('DVBA\*2.7\*189') then

begin

if FMPriorityOfExam.Text = 'INSUFFICIENT EXAM' then

FMPriorityOfExam.Enabled := False

else

FMPriorityOfExam.Enabled := True;

end;

// Added new release status for enabling Insufficient exam button - CodeCR669 JRL 12/15/14

if (Copy(FMRequestStatus.Text + ' ', 1, 9) = 'COMPLETED') OR //COMPLETED, PRINTED BY RO

(Copy(FMRequestStatus.Text + ' ', 1, 14) = 'RELEASED TO RO') then //RELEASED TO RO, NOT PRINTED

begin

btnInsufficientExam.Enabled := True;

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

FMPriorityOfExam.Enabled := False; // This must be here to keep the box off appropriately.

end

else

btnInsufficientExam.Enabled := False;

if IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666/667 JRL 2/5/15 - Added test for backward compatibility

begin

// Replaced Priority of Exam with Claim Type and Special Considerations, CodeCR666/CodeCR667 JRL 11/12/14

if (FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') then // CodeCR666/667 JRL 11/12/14

if Uppercase(FMInsufficientExam.Text) = 'YES' then // CodeCR666/667 JRL 11/12/14

begin // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := FALSE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := FALSE; // CodeCR666/667 JRL 11/12/14

end // CodeCR666/667 JRL 11/12/14

else // CodeCR666/667 JRL 11/12/14

begin // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := TRUE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := TRUE; // CodeCR666/667 JRL 11/12/14

end; // CodeCR666/667 JRL 11/12/14

// Added new release status for enabling Insufficient exam button - CodeCR669 JRL 12/15/14

if (Copy(FMRequestStatus.Text + ' ', 1, 9) = 'COMPLETED') OR //COMPLETED, PRINTED BY RO

(Copy(FMRequestStatus.Text + ' ', 1, 14) = 'RELEASED TO RO') then //RELEASED TO RO, NOT PRINTED

begin // CodeCR666/667 JRL 11/12/14

btnInsufficientExam.Enabled := True; // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := FALSE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := FALSE; // CodeCR666/667 JRL 11/12/14

end // CodeCR666/667 JRL 11/12/14

else // CodeCR666/667 JRL 11/12/14

btnInsufficientExam.Enabled := False; // CodeCR666/667 JRL 11/12/14

end;

end;

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

procedure TfrmViewExam.edExamReferenceNumberChange(Sender: TObject);

begin

Scrollbox1.VertScrollbar.Position := 0;

SaveFlag := False; // Set flags to false

CommentsFlag := False;

if (FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') then

if not IsPatchInstalled('DVBA\*2.7\*189') then

begin

if FMPriorityOfExam.Text = 'INSUFFICIENT EXAM' then

FMPriorityOfExam.Enabled := False

else

FMPriorityOfExam.Enabled := True;

end;

// Added new release status for enabling Insufficient exam button - CodeCR669 JRL 12/15/14

if (Copy(FMRequestStatus.Text + ' ', 1, 9) = 'COMPLETED') OR //COMPLETED, PRINTED BY RO

(Copy(FMRequestStatus.Text + ' ', 1, 14) = 'RELEASED TO RO') then //RELEASED TO RO, NOT PRINTED

begin

btnInsufficientExam.Enabled := True;

if not IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666 JRL 2/3/15 -- Added test for backward compatibility

FMPriorityOfExam.Enabled := False; // This must be here to keep the box off appropriately.

end

else

btnInsufficientExam.Enabled := False;

if IsPatchInstalled('DVBA\*2.7\*189') then // CodeCR666/667 JRL 2/5/15 - Added test for backward compatibility

begin

// Replaced Priority of Exam with Claim Type and Special Considerations, CodeCR666/CodeCR667 JRL 11/12/14

if (FMRequestStatus.Text <> 'NEW') and (FMRequestStatus.Text <> 'PENDING, REPORTED') then // CodeCR666/667 JRL 11/12/14

if Uppercase(FMInsufficientExam.Text) = 'YES' then // CodeCR666/667 JRL 11/12/14

begin // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := FALSE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := FALSE; // CodeCR666/667 JRL 11/12/14

end // CodeCR666/667 JRL 11/12/14

else // CodeCR666/667 JRL 11/12/14

begin // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := TRUE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := TRUE; // CodeCR666/667 JRL 11/12/14

end; // CodeCR666/667 JRL 11/12/14

// Added new release status for enabling Insufficient exam button - CodeCR669 JRL 12/15/14

if (Copy(FMRequestStatus.Text + ' ', 1, 9) = 'COMPLETED') OR //COMPLETED, PRINTED BY RO

(Copy(FMRequestStatus.Text + ' ', 1, 14) = 'RELEASED TO RO') then //RELEASED TO RO, NOT PRINTED

begin // CodeCR666/667 JRL 11/12/14

btnInsufficientExam.Enabled := True; // CodeCR666/667 JRL 11/12/14

cbClaimType.Enabled := FALSE; // CodeCR666 JRL 1/29/15

chklstSpecialConsiderations.Enabled := FALSE; // CodeCR666/667 JRL 11/12/14

end // CodeCR666/667 JRL 11/12/14

else // CodeCR666/667 JRL 11/12/14

btnInsufficientExam.Enabled := False; // CodeCR666/667 JRL 11/12/14

end;

**// If Patch 193 is installed, then determine whether or not the Re-Route**

**// Re-Route Request button should be enabled or disabled. If Patch 193 is**

**// not installed, button will be hidden.**

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**begin // 5/10/16 JRL**

**if CanRequestBeRerouted(edExamReferenceNumber.Text) then //**

**begin //**

**btnRerouteRequest.Visible := TRUE; // If request can be re-routed, //**

**btnRerouteRequest.Enabled := TRUE; // show and enable Re-Route button //**

**end //**

**else //**

**begin //**

**btnRerouteRequest.Visible := TRUE; // If request can't be re-routed, //**

**btnRerouteRequest.Enabled := FALSE;// show but disable Re-Route button //**

**end; //**

**end //**

**else // Patch 192 and earlier, hide Re-Route button //**

**begin //**

**btnRerouteRequest.Visible := FALSE; //**

**btnInsufficientExam.Width := 320; //**

**btnInsufficientExam.Left := 1; //**

**btnClose.Left := 322; //**

**btnClose.Width := 320; //**

**end;**

end;

Table 48: Functions

| Function Name | btnRerouteRequestClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Re-Route request button clicked. Compose request and call RPC to send. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnRerouteRequestClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

**// -----------------------------------------------------------------------------**

**// Re-Route Request button click 5/10/16 JRL Patch 193**

**// -----------------------------------------------------------------------------**

**procedure TfrmViewExam.btnRerouteRequestClick(Sender: TObject);**

**var**

**frmReRouteRequest : TForm;**

**modal\_result : Integer;**

**begin**

**btnRerouteRequest.Enabled := FALSE; // don't allow multiple attempts**

**try**

**// Show Re-Route Request screen**

**frmReRouteRequest := TfrmReRouteRequest.Create(Application);**

**modal\_result := frmReRouteRequest.ShowModal;**

**finally**

**FreeAndNil(frmReRouteRequest);**

**end;**

**// If you submit the rejection, close form. If you cancel the rejection, keep form open**

**if ReturnValue = CLOSE\_FORM then**

**Close**

**else**

**btnRerouteRequest.Enabled := TRUE; // allow re-routing again**

**end;**

Table 48: Functions

| Function Name | FormActivate | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Check for re-routed request and setup form accordingly. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormActivate | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmViewExam.FormActivate(Sender: TObject);

var

x: integer;

flag: boolean;

Patch193 : Boolean; // Patch 193 JRL 8/4/16

ReRoutedRequest : Boolean; // Patch 193 JRL 08/26/16

begin

if uppercase(FMClaimFolderRequiredFake.Text) = 'YES' then

ClaimFolderRequired := True

else

ClaimFolderRequired := False;

if IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 4/17/15

begin // CodeCR704 JRL 4/17/15

if uppercase(FMElectronicClaimFolderFake.Text) = 'YES' then // CodeCR704 JRL 4/17/15

ElectronicClaimFolderAvailable := True // CodeCR704 JRL 4/17/15

else // CodeCR704 JRL 4/17/15

ElectronicClaimFolderAvailable := False; // CodeCR704 JRL 4/17/15

end; // CodeCR704 JRL 4/17/15

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

if uppercase(FMInsufficientExamFake.Text) = 'YES' then // CodeCR669 JRL 11/19/14

FMInsufficientExam.ItemIndex := 1 // YES // CodeCR669 JRL 11/19/14

else // CodeCR669 JRL 11/19/14

FMInsufficientExam.ItemIndex := 0; // NO // CodeCR669 JRL 11/19/14

end;

CurrentRecordIEN := frmMain.FMExamRequestListbox.GetSelectedRecord.IEN;

if FMRegionalOfficeNumber.Text = '' then

FMRegionalOfficeNumber.enabled := True;

btnCancelAllExams.Enabled := False;

if lstExamsRequested.Items.Count > 0 then begin

flag := true;

for x := 0 to lstExamsRequested.Items.Count - 1 do

if Pos('[OPEN]', lstExamsRequested.Items[x]) = 0 then

flag := false;

if flag = true then

btnCancelAllExams.Enabled := True;

end;

// Force update of screen controls

FMRequestStatusChange(Application);

if ReadOnlyMode = True then

btnCancelAllExams.Enabled := False;

if ReadOnlyMode = True then

btnInsufficientExam.Enabled := False;

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

GetClaimType(CurrentRecordIEN); // CodeCR666 JRL 11/12/14

SaveClaimType := FALSE; // CodeCR666 JRL 11/20/14

GetSpecialConsidations(CurrentRecordIEN); // CodeCR667 JRL 11/12/14

SaveSpecialConsiderations := FALSE; // CodeCR667 JRL 11/20/14

end;

// Hide or display re-route fields. If Patch 193, show fields. For

// backward compatibility, hide fields.

st508ReRoutedTo.Visible := Patch193;

edtReRoutedTo.Visible := Patch193;

st508ReRoutedDate.Visible := Patch193;

edtReRoutedDate.Visible := Patch193;

st508ReRoutedStatus.Visible := Patch193;

edtReRoutedStatus.Visible := Patch193;

st508ReRoutedStatusDate.Visible := Patch193;

edtReRoutedStatusDate.Visible := Patch193;

st508ReRoutedFrom.Visible := Patch193;

edtReRoutedFrom.Visible := Patch193;

pageCommentReRoute.ActivePage := tsComments;

tsReRouteInformation.TabVisible := Patch193;

// these fields are only shown if a re-route is done within the same VAMC - enable later if needed

st508RoutingLocTo.Visible := FALSE;

edtReRoutedToRoutingLoc.Visible := FALSE;

st508RoutingLocFrom.Visible := FALSE;

edtReRoutedFromRoutingLoc.Visible := FALSE;

end;

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

procedure TfrmViewExam.FormActivate(Sender: TObject);

var

x: integer;

flag: boolean;

Patch193 : Boolean; // Patch 193 JRL 8/4/16

ReRoutedRequest : Boolean; // Patch 193 JRL 08/26/16

begin

if uppercase(FMClaimFolderRequiredFake.Text) = 'YES' then

ClaimFolderRequired := True

else

ClaimFolderRequired := False;

if IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 4/17/15

begin // CodeCR704 JRL 4/17/15

if uppercase(FMElectronicClaimFolderFake.Text) = 'YES' then // CodeCR704 JRL 4/17/15

ElectronicClaimFolderAvailable := True // CodeCR704 JRL 4/17/15

else // CodeCR704 JRL 4/17/15

ElectronicClaimFolderAvailable := False; // CodeCR704 JRL 4/17/15

end; // CodeCR704 JRL 4/17/15

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

if uppercase(FMInsufficientExamFake.Text) = 'YES' then // CodeCR669 JRL 11/19/14

FMInsufficientExam.ItemIndex := 1 // YES // CodeCR669 JRL 11/19/14

else // CodeCR669 JRL 11/19/14

FMInsufficientExam.ItemIndex := 0; // NO // CodeCR669 JRL 11/19/14

end;

CurrentRecordIEN := frmMain.FMExamRequestListbox.GetSelectedRecord.IEN;

if FMRegionalOfficeNumber.Text = '' then

FMRegionalOfficeNumber.enabled := True;

btnCancelAllExams.Enabled := False;

if lstExamsRequested.Items.Count > 0 then begin

flag := true;

for x := 0 to lstExamsRequested.Items.Count - 1 do

if Pos('[OPEN]', lstExamsRequested.Items[x]) = 0 then

flag := false;

if flag = true then

btnCancelAllExams.Enabled := True;

end;

// Force update of screen controls

FMRequestStatusChange(Application);

if ReadOnlyMode = True then

btnCancelAllExams.Enabled := False;

if ReadOnlyMode = True then

btnInsufficientExam.Enabled := False;

if IsPatchInstalled('DVBA\*2.7\*189') then

begin

GetClaimType(CurrentRecordIEN); // CodeCR666 JRL 11/12/14

SaveClaimType := FALSE; // CodeCR666 JRL 11/20/14

GetSpecialConsidations(CurrentRecordIEN); // CodeCR667 JRL 11/12/14

SaveSpecialConsiderations := FALSE; // CodeCR667 JRL 11/20/14

end;

**// Patch 193 JRL 7/5/16**

**if IsPatchInstalled('DVBA\*2.7\*193') then**

**Patch193 := TRUE**

**else**

**Patch193 := FALSE;**

// Hide or display re-route fields. If Patch 193, show fields. For

// backward compatibility, hide fields.

st508ReRoutedTo.Visible := Patch193;

edtReRoutedTo.Visible := Patch193;

st508ReRoutedDate.Visible := Patch193;

edtReRoutedDate.Visible := Patch193;

st508ReRoutedStatus.Visible := Patch193;

edtReRoutedStatus.Visible := Patch193;

st508ReRoutedStatusDate.Visible := Patch193;

edtReRoutedStatusDate.Visible := Patch193;

st508ReRoutedFrom.Visible := Patch193;

edtReRoutedFrom.Visible := Patch193;

pageCommentReRoute.ActivePage := tsComments;

tsReRouteInformation.TabVisible := Patch193;

// these fields are only shown if a re-route is done within the same VAMC - enable later if needed

st508RoutingLocTo.Visible := FALSE;

edtReRoutedToRoutingLoc.Visible := FALSE;

st508RoutingLocFrom.Visible := FALSE;

edtReRoutedFromRoutingLoc.Visible := FALSE;

**// Complete updating Patch 193 display info // Patch 193 JRL 8/7/16**

**if Patch193 then**

**begin**

**ReRoutedRequest := UpdateReRoutedVistADisplayFields; // Display Re-Route Info if available // Patch 193 JRL 8/4/16**

**// if there is no re-route information, don't display the re-route tab**

**if ReRoutedRequest = FALSE then**

**begin**

**pageCommentReRoute.ActivePage := tsComments;**

**tsReRouteInformation.TabVisible := FALSE;**

**end**

**else // there is re-route information, display tab and control editing**

**begin**

**tsReRouteInformation.TabVisible := TRUE;**

**if ShowAcceptRejectReRouteButtons then // Patch 193 JRL 7/1/16**

**UpdateCommentsReRouteTabs(TAB\_REROUTE\_ACCEPT\_REJECT\_MODE)**

**else**

**UpdateCommentsReRouteTabs(TAB\_REROUTE\_DISPLAY\_INFO\_MODE);**

**end;**

**end; // if Patch193**

end;

Table 48: Functions

| Function Name | btnAcceptReRouteClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Accept Re-Routed Request button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnAcceptReRouteClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Accept Re-Route Request button click 7/18/16 JRL Patch 193

// -----------------------------------------------------------------------------

procedure TfrmViewExam.btnAcceptReRouteClick(Sender: TObject);

const

SUCCESS = TRUE;

begin

if SendAcceptReRouteInfo = SUCCESS then

begin

UpdateCommentsReRouteTabs(TAB\_REROUTE\_DISPLAY\_INFO\_MODE);

frmViewExam.FMGetsExamRequest.GetAndFill; // refresh fields including request status

UpdateReRoutedVistADisplayFields; // refresh VistA Re-Route fields

SaveFlag := FALSE; // reset form update flag so screen refreshes don't trigger a save message

end;

end;

Table 48: Functions

| Function Name | btnRejectReRouteClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Reject Re-Routed Request button clicked | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnRejectReRouteClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Reject Re-Route Request button click 7/18/16 JRL Patch 193

// -----------------------------------------------------------------------------

procedure TfrmViewExam.btnRejectReRouteClick(Sender: TObject);

var

frmReRouteReject : TForm;

begin

try

// Show Rejection screen

frmReRouteReject := TfrmReRouteReject.Create(Application);

frmReRouteReject.ShowModal;

// If you submit the rejection, close form. If you cancel the rejection, keep form open

if ReturnValue = CLOSE\_FORM then

frmViewExam.Close;

finally

FreeAndNil(frmReRouteReject);

end;

end;

Table 48: Functions

| Function Name | CanRequestBeRerouted | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Call RPC to see if request can be re-routed. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | CanRequestBeRerouted | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Can Request Be Re-Routed? 5/10/16 JRL Patch 193

// -----------------------------------------------------------------------------

// A Request can be re-routed if it has one of the statuses below and hasn't

// already been re-routed.

// REQUEST STATUS:

// \* NEW

// \* PENDING, REPORTED

// \* NEW, RE-ROUTED

// -----------------------------------------------------------------------------

// Input: Request IEN

// Output: '0' - request cannot be re-routed

// '1' - request can be re-routed

// ^

// Current Station Number (needed for re-routing within the current site)

// -----------------------------------------------------------------------------

function TfrmViewExam.CanRequestBeRerouted(RequestIEN: String): Boolean;

CONST

RPCNAME = 'DVBA CAPRI GET REROUTE';

var

i : Integer;

Status : string;

begin

// If the user is logged in remotely, the re-routing is disabled

// Re-routing is only available to local users

if ESSOVersion then // ESSOVersion = TRUE indicates remotely logged in user

begin

Result := FALSE;

exit;

end;

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].Value := RequestIEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On E: EBrokerError Do

ShowMessageCAPRI(RPCNAME + ' could not be accessed!'+#13#10+E.Message);

End;

Result := FALSE; // Request cannot be re-routed

If frmMain.RPCBroker1.Results.Count > 0 Then

begin

Status := Piece(frmMain.RPCBroker1.Results[0], '^', 1);

CurrentStationNumber := Piece(frmMain.RPCBroker1.Results[0], '^', 2); // store station number for possible re-routing later

if Status = '1' then // Request can be re-routed

Result := TRUE;

end;

end;

Table 48: Functions

| Function Name | UpdateCommentsReRouteTabs | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | If the request is a newly re-routed request, we want to display the  Re-Route tab and Re-Route buttons. Highlight tab in orange to make it stand out. All controls should be read-only until the request is accepted/rejected. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | UpdateCommentsReRouteTabs | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Update Comments / Re-Route Tab Patch 193 JRL 8/7/16

// -----------------------------------------------------------------------------

// If the request is a newly re-routed request, we want to display the

// Re-Route tab and Re-Route buttons. Highlight tab in orange to make it stand

// out. All controls should be read-only until the request is accepted/rejected.

// Input Parameter is: TAB\_REROUTE\_ACCEPT\_REJECT\_MODE

//

// If the request has already been accepted, the re-route tab should be blue.

// All controls are editable.

// If the re-route status is shown as accepted, then show the comments tab on top.

// If the re-route status is shown as rejected and the current request status is

// NEW or PENDING, REPORTED, show the re-route tab on top. This enables a user

// to see why a request has been rejected.

// Input Parameter is: TAB\_REROUTE\_DISPLAY\_INFO\_MODE

// -----------------------------------------------------------------------------

procedure TfrmViewExam.UpdateCommentsReRouteTabs(Tab\_ReRoute: String);

const

REROUTED\_REJECTED = 'REJECTED';

NEW = 'NEW';

PENDING = 'PENDING, REPORTED';

PENDING2 = 'PENDING, SCHEDULED';

NEW\_REROUTED = 'NEW, RE-ROUTED';

OPEN = 'OPEN';

ORANGE = $000080FF;

BLUE = $00FBE2AE;

begin

if Tab\_ReRoute = TAB\_REROUTE\_ACCEPT\_REJECT\_MODE then

begin

UpdateControlsForEditingOrReadOnly(FALSE); // Read Only Mode until you accept

pageCommentReRoute.ActivePage := tsReRouteInformation; // Re-Route Tab

btnAcceptReRoute.Visible := TRUE; // Show Accept button

btnRejectReRoute.Visible := TRUE; // Show Reject button

pnlReRouteButtons.Color := ORANGE; // panel color orange

st508ReRouteReason.Color := ORANGE; // text background orange

st508ReRouteDescription.Color := ORANGE; // text background orange

end

else // Tab\_ReRoute = TAB\_REROUTE\_DISPLAY\_INFO\_MODE

begin

// check statuses to determine if fields should be editable.

if (FMRequestStatus.Text = NEW) OR

(FMRequestStatus.TEXT = OPEN) OR

(FMRequestStatus.Text = PENDING) OR

(FMRequestStatus.Text = PENDING2) OR

(FMRequestStatus.Text = REROUTED\_REJECTED) then

UpdateControlsForEditingOrReadOnly(TRUE) // Edit Mode

else

UpdateControlsForEditingOrReadOnly(FALSE); // Read-Only Mode

pageCommentReRoute.ActivePage := tsComments; // Comment Tab

if edtReRoutedStatus.Text = REROUTED\_REJECTED then // rejected, always show

pageCommentReRoute.ActivePage := tsReRouteInformation; // rejection tab

btnAcceptReRoute.Visible := FALSE; // Hide Accept button

btnRejectReRoute.Visible := FALSE; // Hide Reject button

pnlReRouteButtons.Color := BLUE; // panel color blue

st508ReRouteReason.Color := BLUE; // text background blue

st508ReRouteDescription.Color := BLUE; // text background blue

st508RejectReason.Color := BLUE; // text background blue

end;

end;

Table 48: Functions

| Function Name | UpdateControlsForEditingOrReadOnly | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Make controls read-only or editable based on re-route status. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | UpdateControlsForEditingOrReadOnly | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Update Controls for Editing or Read Only Patch 193 JRL 7/5/16

// -----------------------------------------------------------------------------

// Do not allow the user to edit controls until they have accepted the

// re-routed request. This routine enabled or disables the controls

// accordingly.

// -----------------------------------------------------------------------------

procedure TfrmViewExam.UpdateControlsForEditingOrReadOnly(Enabled: Boolean);

begin

fmCommentsMemo.ReadOnly := not Enabled;

btnAddExam.Enabled := Enabled;

fmOtherDisabilities1.Enabled := Enabled;

fmOtherDisabilities2.Enabled := Enabled;

fmOtherDisabilities3.Enabled := Enabled;

fmRoutingLocation.Enabled := Enabled;

fmRequestedBy.Enabled := Enabled;

cbClaimType.Enabled := Enabled;

chklstSpecialConsiderations.Enabled := Enabled;

fmOtherDocumentsRequired.Enabled := Enabled;

fmClaimFolderRequired.Enabled := Enabled;

fmElectronicClaimFolder.Enabled := Enabled;

FMRegionalOfficeNumber.Enabled := Enabled;

fmRoutingLocation.Enabled := Enabled;

fmRegionalOfficeNumber.Enabled := Enabled;

dtpLastRatingExamDate.Enabled := Enabled;

fmLastRatingExamDate.Enabled := Enabled;

end;

Table 48: Functions

| Function Name | UpdateReRoutedVistADisplayFields | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Check for re-routed request and setup form accordingly. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | UpdateReRoutedVistADisplayFields | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Update VistA Display Fields for Re-Routed Information Patch 193 JRL 8/4/16

// -----------------------------------------------------------------------------

// RPC: DVBA CAPRI REROUTE INFO

// Input: Request IEN

// Output:

// Line 1: Re-Route To Site ^

// Re-Route Date ^

// Re-Route Status ^

// Re-Route Status Date ^

// Re-Routed From Site ^

// Re-Route Reason ^

// Show/Hide Accept and Reject buttons (1 = show) ^

// Re-Route From Routing Location ^

// Re-Route To Routing Location

// Line 2: Re-Route Description (250 chars)

// Line 3: Rejection Reason (250 chars)

//

// Procedure returns TRUE if the request has re-route information, FALSE otherwise

//------------------------------------------------------------------------------

function TfrmViewExam.UpdateReRoutedVistADisplayFields : Boolean;

const

RPCNAME = 'DVBA CAPRI REROUTE INFO';

NOT\_REROUTED = 'THIS 2507 REQUEST HAS NOT BEEN REROUTED';

REROUTED\_NEW = 'NEW';

REROUTED\_ACCEPTED = 'ACCEPTED';

REROUTED\_REJECTED = 'REJECTED';

var

ReRoutedMsg : String;

i : Integer;

Flag : String;

begin

// clear all the fields

edtReRoutedTo.Clear;

edtReRoutedDate.Clear;

edtReRoutedStatus.Clear;

edtReRoutedStatusDate.Clear;

edtReRoutedFrom.Clear;

edtReRouteReason.Clear;

edtReRoutedToRoutingLoc.Clear;

edtReRoutedFromRoutingLoc.Clear;

memoReRouteDescription.Lines.Clear;

memoRejectReason.Lines.Clear;

// Get Re-Route Information

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := edExamReferenceNumber.Text;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

On E: EBrokerError Do

ShowMessageCAPRI('RPC: '+RPCNAME + ' could not be accessed! '+ e.Message);

end;

if frmMain.RPCBroker1.Results.Count > 0 then

begin

// RPC returns the message: THIS 2507 REQUEST HAS NOT BEEN REROUTED

// in the 'Re-Routed To' field if this request has no re-routing data.

// Do not display that message.

ReRoutedMsg := Uppercase(Piece(frmMain.RPCBroker1.Results[0], '^', 1));

if ReRoutedMsg <> NOT\_REROUTED then // valid data, fill in display

begin

edtReRoutedTo.Text := ReRoutedMsg;

edtReRoutedDate.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 2);

edtReRoutedStatus.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 3);

edtReRoutedStatusDate.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 4);

edtReRoutedFrom.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 5);

edtReRouteReason.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 6);

Flag := Piece(frmMain.RPCBroker1.Results[0], '^', 7); // re-route button flag

edtReRoutedFromRoutingLoc.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 8);

edtReRoutedToRoutingLoc.Text := Piece(frmMain.RPCBroker1.Results[0], '^', 9);

// if re-routed from matches re-routed to, we are re-routing locally to the

// same VAMC. Display Routing location information and set "local reroute" flag

LocalReRoute := FALSE;

if edtReRoutedTo.Text = edtReRoutedFrom.Text then

begin

st508RoutingLocFrom.Visible := TRUE;

edtReRoutedFromRoutingLoc.Visible := TRUE;

st508RoutingLocTo.Visible := TRUE;

edtReRoutedToRoutingLoc.Visible := TRUE;

LocalReRoute := TRUE;

end;

// check whether or not re-route buttons should be displayed

if Flag = '1' then

ShowAcceptRejectReRouteButtons := TRUE

else

ShowAcceptRejectReRouteButtons := FALSE;

// setup Re-Route tab display and data

if (edtReRoutedStatus.Text = REROUTED\_ACCEPTED) OR

(edtReRoutedStatus.Text = REROUTED\_NEW) then // no reject reason displayed

begin

memoReRouteDescription.Height := 85;

st508RejectReason.Visible := FALSE;

memoRejectReason.Visible := FALSE;

if frmMain.RPCBroker1.Results.Count > 1 then

memoReRouteDescription.Lines.Add(frmMain.RPCBroker1.Results[1]);

end;

if (edtReRoutedStatus.Text = REROUTED\_REJECTED) then

begin

memoReRouteDescription.Height := 35;

st508RejectReason.Visible := TRUE;

memoRejectReason.Visible := TRUE;

if frmMain.RPCBroker1.Results.Count > 1 then

memoReRouteDescription.Lines.Add(frmMain.RPCBroker1.Results[1]);

if (frmMain.RPCBroker1.Results.Count > 2) then

memoRejectReason.Lines.Add(frmMain.RPCBroker1.Results[2]);

end;

Result := TRUE; // re-route information exists

end

else

Result := FALSE; // no re-route information exists

end;

end;

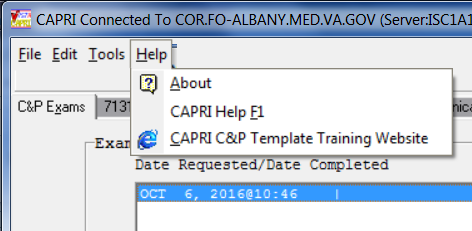
##### GUI – Main.pas

Table 36: GUI

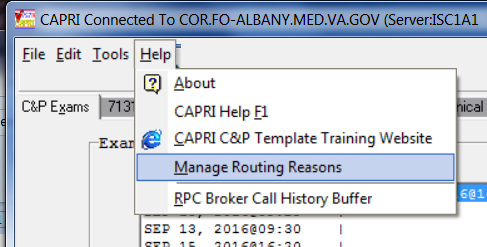
| Unit Name | Description |
| --- | --- |
| Main.pas | Modified forms to setup menu items and inactivate routing locations. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form



##### Modified Form



##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| ActHelpEdtReRouteReasons | TAction | “Manage Routing Reasons” menu option |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| actToolsExamListParametersExecute | Menu Item clicked | Modified to set any newly identified routing location and set them to inactive instead of active. |
| actFileConnectExecute | Menu Item Clicked | Modified to check for security key and whether or not to display menu item. |
| btnAddRequestClick | Button Clicked | Modified to update Request Status. |

##### Methods – N/A for Requirement 2.6.1

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause

EdtRerouteReasons

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | main | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** | Main CAPRI form | | | |

| Current Form Layout |
| --- |
| See current form above |

| Modified Form Layout (Changes are in bold) |
| --- |
| See modified form above |

##### Functions

Table 48: Functions

| Function Name | actToolsExamListParametersExecute | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to set any newly identified routing location and set them to inactive instead of active. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | actToolsExamListParametersExecute | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmMain.actToolsExamListParametersExecute(Sender: TObject);

Var

x, y, z: integer;

xx, yy, zz: integer;

foundflag: boolean;

tempSites: TStringList;

IEN, ZIEN: String;

Begin

If ANURemoteProcedureCallInProgress = True Then

exit;

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating exam assignment dialog.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

tempSites := TStringList.Create;

tempSites.Clear;

frmSiteExamList := TfrmSiteExamList.Create(frmMain);

//Set up exams list

frmSiteExamList.FMExamsList.Items := ListExams;

If frmSiteExamList.FMExamsList.Items.Count > 0 Then

For x := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

If Pos(' I', frmSiteExamList.FMExamsList.Items[x]) > 0 Then

frmSiteExamList.FMExamsList.Items.Delete(x);

// Make sure CAPRI DIVISION EXAM LIST FILE contains an entry in .01 for each division

// The site may have added a new division.

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := FMListerMedicalCenterDivision;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count = 0 Then

Begin

ShowMessageCAPRI('No divisions can be located in VistA. Exiting this function.');

frmSiteExamList.Release;

frmSiteExamList := Nil;

TempHSMemo.Free;

exit;

End;

// Get current list of divisions

For x := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

tempSites.Add(frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN);

End;

// Now get list of divisions defined in the CAPRI DIVISION EXAM LIST FILE

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := frmSiteExamList.FMListerCustomExamListDivisions;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Find IEN and show that in the list

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

frmSiteExamList.CustomExamListDivisions.IENS := frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmSiteExamList.CustomExamListDivisions.GetData;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] := frmSiteExamList.CustomExamListDivisions.GetField('.01').FMDBInternal;

End;

// Compare the two lists and add medical center divisions that are missing

For x := 0 To tempSites.Count - 1 Do

Begin

foundflag := false;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

If foundflag = false Then

Begin

If tempSites[x] = frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] Then

Begin

foundflag := true;

End;

End;

End;

If FoundFlag = False Then

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := '`' + tempSites[x];

frmSiteExamList.FMEditCAPRIDivisionExamList.IENS := '+1,';

If frmSiteExamList.FMEditCAPRIDivisionExamList.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIDivisionExamList);

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMTag;

exit;

End;

If Not frmMain.FMValidator1.Validate Then

Begin

FMFiler1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating new divisions.');

exit;

End;

IEN := FMFiler1.FindIEN('+1,') + ',';

FMFiler1.AddFDA('396.151', '+1,' + IEN, '.01', 'DEFAULT LIST');

FMFiler1.AddFDA('396.151', '+1,' + IEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

ZIEN := FMFiler1.FindIEN('+1,') + ',' + IEN;

For z := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x] + ', Exam: ' + frmSiteExamList.FMExamsList.Items[z];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMExamsList.ItemIndex := z;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := '`' + frmSiteExamList.FMExamsList.GetSelectedRecord.IEN;

frmSiteExamList.FMEditCAPRIExamListDefinition.IENS := '+1,' + ZIEN;

If frmSiteExamList.FMEditCAPRIExamListDefinition.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIExamListDefinition);

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMTag;

exit;

End;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

If Not frmMain.FMValidator1.Validate Then

Begin

FMValidator1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating the exam list.');

exit;

End;

End; // end of exam list population

End;

End;

//Reload division list in case it changed

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Now evaluate for exams that have been inactivated or exams that have been added

StatusBarLoadPt.Caption := 'Validating list consistency.';

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

// Get list for division

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

// Load up the list and see what exams are in it.

If frmSiteExamList.FMListBoxAvailableExamLists.Items.Count > 0 Then

frmSiteExamList.FMListerExamSetup.ListerFlags := [lfInternal];

For z := 0 To frmSiteExamList.FMListBoxAvailableExamLists.Items.Count - 1 Do

Begin

frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex := z;

StatusBarLoadPt.Caption := 'Validating ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMListBoxAvailableExamLists.Items[frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex];

frmSiteExamList.buttonEditClick(Application);

// Check for exams now inactive

For xx := frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMListBoxExamSetup.ItemIndex := xx;

foundflag := false;

For yy := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMExamsList.ItemIndex := yy;

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[xx], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[xx]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

StatusBarLoadPt.Caption := 'Removing ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.CheckListBoxExamList.Items[xx];

frmSiteExamList.FMEditExamNameInList.Text := '@';

frmSiteExamList.FMEditExamNameInList.IENS := frmSiteExamList.FMListBoxExamSetup.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' +

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

If frmSiteExamList.FMEditExamNameInList.Validate Then

Begin

frmMain.FMFiler1.AddChgdControl(frmSiteExamList.FMEditExamNameInList);

frmSiteExamList.FMEditExamNameInList.Text := frmSiteExamList.FMEditExamNameInList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

exit;

End;

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

For xx := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

foundflag := false;

frmSiteExamList.FMExamsList.ItemIndex := xx;

For yy := 0 To frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Do

Begin

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[yy], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[yy]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

//ShowMessageCAPRI('turn on '+frmSiteExamList.FMExamsList.Items[xx]);

StatusBarLoadPt.Caption := 'Adding ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMExamsList.Items[xx];

ZIEN := frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '.01', frmSiteExamList.FMExamsList.GetSelectedRecord.IEN);

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

End;

frmSiteExamList.FMListerExamSetup.ListerFlags := [];

End;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := 0;

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

frmSiteExamList.top := frmMain.Top + ((frmMain.Height - frmSiteExamList.Height) Div 2);

frmSiteExamList.left := frmMain.left + ((frmMain.Width - frmSiteExamList.Width) Div 2);

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.Panel1.BringToFront;

// Set Up Institutions

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmSiteExamList.FMListBoxUserGroup.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

frmSiteExamList.FMMemoDivisionComments.Enabled := True;

ContextorChangeMessage := 'You are editing site parameters. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

frmSiteExamList.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmSiteExamList.Release;

frmSiteExamList := Nil;

tempSites.Free;

end;

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

procedure TfrmMain.actToolsExamListParametersExecute(Sender: TObject);

Var

x, y, z: integer;

xx, yy, zz: integer;

foundflag: boolean;

tempSites: TStringList;

IEN, ZIEN: String;

Begin

If ANURemoteProcedureCallInProgress = True Then

exit;

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating exam assignment dialog.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

tempSites := TStringList.Create;

tempSites.Clear;

frmSiteExamList := TfrmSiteExamList.Create(frmMain);

//Set up exams list

frmSiteExamList.FMExamsList.Items := ListExams;

If frmSiteExamList.FMExamsList.Items.Count > 0 Then

For x := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

If Pos(' I', frmSiteExamList.FMExamsList.Items[x]) > 0 Then

frmSiteExamList.FMExamsList.Items.Delete(x);

// Make sure CAPRI DIVISION EXAM LIST FILE contains an entry in .01 for each division

// The site may have added a new division.

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := FMListerMedicalCenterDivision;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count = 0 Then

Begin

ShowMessageCAPRI('No divisions can be located in VistA. Exiting this function.');

frmSiteExamList.Release;

frmSiteExamList := Nil;

TempHSMemo.Free;

exit;

End;

// Get current list of divisions

For x := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

tempSites.Add(frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN);

End;

// Now get list of divisions defined in the CAPRI DIVISION EXAM LIST FILE

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := frmSiteExamList.FMListerCustomExamListDivisions;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Find IEN and show that in the list

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

frmSiteExamList.CustomExamListDivisions.IENS := frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmSiteExamList.CustomExamListDivisions.GetData;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] := frmSiteExamList.CustomExamListDivisions.GetField('.01').FMDBInternal;

End;

// Compare the two lists and add medical center divisions that are missing

For x := 0 To tempSites.Count - 1 Do

Begin

foundflag := false;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

If foundflag = false Then

Begin

If tempSites[x] = frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] Then

Begin

foundflag := true;

End;

End;

End;

If FoundFlag = False Then

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := '`' + tempSites[x];

frmSiteExamList.FMEditCAPRIDivisionExamList.IENS := '+1,';

If frmSiteExamList.FMEditCAPRIDivisionExamList.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIDivisionExamList);

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMTag;

exit;

End;

If Not frmMain.FMValidator1.Validate Then

Begin

FMFiler1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating new divisions.');

exit;

End;

IEN := FMFiler1.FindIEN('+1,') + ',';

**// Patch 193 - New divisions should be set to inactive instead of active.**

**// This keeps VBA users from erroneously sending requests to new divisions**

**// that are not prepared to receive them. Initially Remedy INC000001290246**

**// but not a defect. Incorporated into Patch 193 requirements.**

**// Patch 193 RRA 7/18/16**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 RRA 7/18/16**

**begin // Patch 193 RRA 7/18/16**

**FMFiler1.AddFDA('396.15', IEN, '4', 'Y'); // Patch 193 RRA 7/18/16**

**if not FMFiler1.Update then // Patch 193 RRA 7/18/16**

**begin // Patch 193 RRA 7/18/16**

**ShowMessageCAPRI('There was a problem updating new divisions.'); // Patch 193 RRA 7/18/16**

**exit; // Patch 193 RRA 7/18/16**

**end; // Patch 193 RRA 7/18/16**

**end; // Patch 193 RRA 7/18/16**

**// end RRA**

FMFiler1.AddFDA('396.151', '+1,' + IEN, '.01', 'DEFAULT LIST');

FMFiler1.AddFDA('396.151', '+1,' + IEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

ZIEN := FMFiler1.FindIEN('+1,') + ',' + IEN;

For z := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x] + ', Exam: ' + frmSiteExamList.FMExamsList.Items[z];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMExamsList.ItemIndex := z;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := '`' + frmSiteExamList.FMExamsList.GetSelectedRecord.IEN;

frmSiteExamList.FMEditCAPRIExamListDefinition.IENS := '+1,' + ZIEN;

If frmSiteExamList.FMEditCAPRIExamListDefinition.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIExamListDefinition);

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMTag;

exit;

End;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

If Not frmMain.FMValidator1.Validate Then

Begin

FMValidator1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating the exam list.');

exit;

End;

End; // end of exam list population

End;

End;

//Reload division list in case it changed

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Now evaluate for exams that have been inactivated or exams that have been added

StatusBarLoadPt.Caption := 'Validating list consistency.';

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

// Get list for division

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

// Load up the list and see what exams are in it.

If frmSiteExamList.FMListBoxAvailableExamLists.Items.Count > 0 Then

frmSiteExamList.FMListerExamSetup.ListerFlags := [lfInternal];

For z := 0 To frmSiteExamList.FMListBoxAvailableExamLists.Items.Count - 1 Do

Begin

frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex := z;

StatusBarLoadPt.Caption := 'Validating ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMListBoxAvailableExamLists.Items[frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex];

frmSiteExamList.buttonEditClick(Application);

// Check for exams now inactive

For xx := frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMListBoxExamSetup.ItemIndex := xx;

foundflag := false;

For yy := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMExamsList.ItemIndex := yy;

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[xx], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[xx]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

StatusBarLoadPt.Caption := 'Removing ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.CheckListBoxExamList.Items[xx];

frmSiteExamList.FMEditExamNameInList.Text := '@';

frmSiteExamList.FMEditExamNameInList.IENS := frmSiteExamList.FMListBoxExamSetup.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' +

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

If frmSiteExamList.FMEditExamNameInList.Validate Then

Begin

frmMain.FMFiler1.AddChgdControl(frmSiteExamList.FMEditExamNameInList);

frmSiteExamList.FMEditExamNameInList.Text := frmSiteExamList.FMEditExamNameInList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

exit;

End;

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

For xx := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

foundflag := false;

frmSiteExamList.FMExamsList.ItemIndex := xx;

For yy := 0 To frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Do

Begin

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[yy], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[yy]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

//ShowMessageCAPRI('turn on '+frmSiteExamList.FMExamsList.Items[xx]);

StatusBarLoadPt.Caption := 'Adding ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMExamsList.Items[xx];

ZIEN := frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '.01', frmSiteExamList.FMExamsList.GetSelectedRecord.IEN);

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

End;

frmSiteExamList.FMListerExamSetup.ListerFlags := [];

End;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := 0;

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

frmSiteExamList.top := frmMain.Top + ((frmMain.Height - frmSiteExamList.Height) Div 2);

frmSiteExamList.left := frmMain.left + ((frmMain.Width - frmSiteExamList.Width) Div 2);

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.Panel1.BringToFront;

// Set Up Institutions

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmSiteExamList.FMListBoxUserGroup.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

frmSiteExamList.FMMemoDivisionComments.Enabled := True;

ContextorChangeMessage := 'You are editing site parameters. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

frmSiteExamList.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmSiteExamList.Release;

frmSiteExamList := Nil;

tempSites.Free;

end;

Table 48: Functions

| Function Name | actFileConnectExecute | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to check for security key and whether or not to display menu item. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | actFileConnectExecute | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmMain.actFileConnectExecute(Sender: TObject);

Var

boReviewerKey: Boolean;

Flag: Boolean;

FoundCPWMKey: String;

IsOldVersionAllowed: Boolean;

ManifestCnt: Integer; // rpk 7/2/2014

tempflag: boolean;

TempPort: Integer;

TempSender: TAction;

TempServer: String;

tempstring: String;

TempString2: String;

TempStrPort: String;

TempStrServer: String;

versiona: real;

versionb: real;

versionrevisionst: String;

versionshown: boolean;

VistAVersion: String;

x: integer;

xversiona: real;

xversionb: real;

xversionrevisionst: String;

xx: integer;

yy: integer;

zz: integer;

tempClientItem: TActionClientItem;

aStream: TMemoryStream;

authToken: string;

VDUrl: string;

edtstr: String;

actnItemSeparator: TActionClientItem; // CodeCR668 rpk 1/21/2015

Begin

// Code "1892890Dhdoy\*(@hDH28y(#\*yUHKJDH"

If ANURemoteProcedureCallInProgress = True Then

exit;

If CCOWMode = False Then

Try

If RPCBroker1.Contextor <> Nil Then

RPCBroker1.Contextor := Nil;

Except

End;

If Essoversion = false Then

labelDivision.visible := False;

If Assigned(formTIUDisplay) Then

Begin

Try

formTIUDisplay.Visible := False;

formTIUDisplay.Release;

formTIUDisplay := Nil;

Except

End;

End;

If (actFileConnect.caption = '&Disconnect...') And

Assigned(PNCSForm) Then

Begin

ShowMessage('Please close the open Template.');

PNCSForm.Show;

Exit;

End;

//AnimateLogo(True);

TempSender := Nil;

If Uppercase(Sender.Classname) = 'TACTION' Then

Begin

TempSender := Sender As TAction;

End;

versionshown := false;

If (actFileConnect.Caption = '&Connect...') Or ((Assigned(TempSender)) And (TempSender = actFileSwitchSites)) Then

Begin //connect

// Dec 2005 Chuck Sener Reports that the GUI sometimes hangs at this spot and never

// shuts down

// Put in a 60 second timer and close CAPRI if no connection in that time.

TimerInitialConnection.Enabled := True;

If SwitchToSite = '' Then

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShuttingDown := True;

timerhalt.enabled := true;

application.terminate;

exit;

End;

// If it gets to this spot, the connection worked, so turn off the timer.

TimerInitialConnection.Enabled := False;

If SwitchToSite = '' Then

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

//timerhalt.enabled:=true;

application.terminate;

exit;

End;

If formESSOSelect = Nil Then

Begin

formESSOSelect := TformESSOSelect.Create(frmMain);

HomeServer := RPCBroker1.Server;

HomePort := InttoStr(RPCBroker1.ListenerPort);

//CodeCR353 - rpm 5/14/12

//Populate Virtual VA URL value on initial connection to home server

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['6'], nil) then

begin

if RPCBroker1.Results.Count > 0 then

begin

FVirtualVAURL := RPCBroker1.Results[0];

//CodeCR353 - rpm 5/31/12

//A URL result indicates that we can populate the Virtual VA TOKEN

//value because the server patch is loaded at the site.

if (FVirtualVAURL <> '') then

begin

if CallRPC(RPCBroker1, 'DVBAB GET VVA TOKEN', [], nil) then

if RPCBroker1.Results.Count > 0 then

FVirtualVAToken := RPCBroker1.Results[0];

end;

end;

end;

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['7'], nil) then // CodeCR440 (AMIS290) JRL 11/15/12

begin // CodeCR440 (AMIS290) JRL 11/15/12

if RPCBroker1.Results.Count > 0 then // CodeCR440 (AMIS290) JRL 11/15/12

begin // CodeCR440 (AMIS290) JRL 11/15/12

FAMIS290URL := RPCBroker1.Results[0]; // CodeCR440 (AMIS290) JRL 11/15/12

//If an URL exists, this indicates that the AMIS290 // CodeCR440 (AMIS290) JRL 11/15/12

//reports have been replaced. New AMIS290 functionality // CodeCR440 (AMIS290) JRL 11/15/12

//is found at the URL location // CodeCR440 (AMIS290) JRL 11/15/12

end; // CodeCR440 (AMIS290) JRL 11/15/12

end; // CodeCR440 (AMIS290) JRL 11/15/12

if TIdHTTPVler.GetVlerDasURL() = '' then //could have been set via command line param //CodeCR500 JRL 9/10/13

begin //CodeCR500 JRL 9/10/13

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['8'], nil) then //CodeCR500 JRL 08/27/13

begin //CodeCR500 JRL 08/27/13

if RPCBroker1.Results.Count > 0 then //CodeCR500 JRL 08/27/13

begin //CodeCR500 JRL 08/27/13

TIdHTTPVler.SetVlerDasURL(RPCBroker1.Results[0]); //CodeCR??? LMS 2014-07-21 Moving from main form field. Needs to be available App wide in all units that need VLER access

end; //CodeCR500 JRL 08/27/13

end; //CodeCR500 JRL 08/27/13

end;

End;

TempServer := RPCBroker1.Server;

TempPort := RPCBroker1.ListenerPort;

SetFont;

If frmMailMessage = Nil Then

frmMailMessage := TfrmMailMessage.Create(frmMain);

If ESSOVersion = True Then

Begin

TimeOutTimer.Enabled := True;

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DTIME'], nil, True) then

exit;

If TimeOutVal < 0 Then // Need this to keep value from getting reset when switching sites

TimeOutVal := StrToInt(RPCBroker1.Results[0]);

If TimeOutVal < 600 Then

TimeOutVal := 600;

If frmMailMan = Nil Then

Begin

// Only check mail on home server once at start-up

frmMailMan := TfrmMailMan.Create(frmMain);

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010 - Why are we making this call?

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

If frmMailMan = Nil Then

frmMailMan := TfrmMailMan.Create(frmMain);

setfont;

End;

setfont;

formEssoSelect.StaticText1.Caption := 'Select a site.';

ContextorChangeMessage := 'The application is in the process of switching VistA accounts. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

FormEssoSelect.formstyle := fsStayOnTop;

If FormEssoSelect.Showmodal = mrcancel Then

Begin // Check for cancel button and exit

If ESSOInitialized = True Then

If RPCBroker1.Connected = True Then

exit;

ShuttingDown := True;

ContextorChangeMessage := '';

CCOWBreakLink := False;

actFileQuitExecute(Application);

End;

// CodeCR185 -MER 04/2011

if Assigned(ReflectionWrapper) then

ReflectionWrapper.TerminateEmulator;

// Drop telnet connection if one exists

If assigned(frmTelnet) Then

If frmTelnet.bitbtnDisconnect.Visible = True Then

Begin

frmTelnet.DisconnectBtnClick(Self);

End;

ESSOConnecting := True;

ESSOInitialized := True;

If (TempServer = RPCBroker1.Server) And (TempPort = RPCBroker1.ListenerPort) Then

Begin

exit;

End;

TimerRemoteData.Enabled := False;

ListBoxRemoteDataPending.Items.Clear;

If frmMain.RPCBroker1.Server = '' Then

Begin

application.terminate;

End;

tempClientItem := ActionManMain.FindItemByCaption('Ex&ternal Applications');

If Assigned(tempClientItem) Then

tempClientItem.Visible := True;

//Set HIA Search menu Item under Tools menu to enable true

// MNJ - 09/10/2012

actHIASearch.Enabled := True;

End

else

begin

tempClientItem := ActionManMain.FindItemByCaption('Ex&ternal Applications');

If Assigned(tempClientItem) Then

tempClientItem.Visible := False;

end;

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

ESSOConnecting := False;

If RPCBroker1.Connected = True Then

Begin

TimeOutTimer.Enabled := True;

PatientIEN := '';

patientIENDOD := '';

frmMain.Caption := ProgramNameCaption;

ClearForm; //CodeCR248 - rpm 3/8/12 - clear the header and tab set

If CCOWMode = True Then

Begin

//If ESSOVersion<>True then begin

// Get local station number

RPCBroker1.RemoteProcedure := 'DVBAB CCOW';

RPCBroker1.Param[0].Value := '1';

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'DVBAB CCOW could not be accessed!';

ProgressBarLoadPt.Repaint;

frmMain.Repaint;

ShowMessageCAPRI('DVBAB CCOW could not be accessed!');

application.terminate;

application.terminate;

End;

End;

LocalStationNumber := Piece(RPCBroker1.Results[0], '^', 3);

// Get Test system status

RPCBroker1.RemoteProcedure := 'DVBAB CCOW';

RPCBroker1.Param[0].Value := '2';

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'DVBAB CCOW could not be accessed!';

ProgressBarLoadPt.Repaint;

frmMain.Repaint;

ShowMessageCAPRI('DVBAB CCOW could not be accessed!');

application.terminate;

application.terminate;

End;

End;

TestSystemLogin := RPCBroker1.Results[0];

//End;

End;

{Reload Divisions and Institutions on New System}

frmSPlashScreen := TfrmSplashScreen.Create(frmMain);

frmSplashScreen.PanelVersion.Height := Panel1.Height;

frmSplashScreen.PanelVersion.Caption := VersionUser;

frmSplashScreen.PanelVersion.Left := (frmSPlashScreen.Width Div 2) - (frmSPlashScreen.PanelVersion.Width Div 2);

frmSPlashScreen.StatusLabel.Height := Panel1.Height;

frmSPlashScreen.Shape1.Height := Panel1.Height;

frmSPlashScreen.GaugeLoadSite.Progress := 0;

frmSPlashScreen.GaugeLoadSite.ForeColor := clRed;

frmSPlashScreen.GaugeLoadSite.Top := frmSPlashScreen.Shape1.Top + frmSPlashScreen.Shape1.Height;

frmSPlashScreen.Height := frmSPlashScreen.GaugeLoadSite.Top + 32;

frmSPlashScreen.Panel1.Height := frmSPlashScreen.GaugeLoadSite.Top + 33;

frmSPlashScreen.StatusLabel.Font := panel1.font;

frmSplashScreen.Show;

frmSplashScreen.Repaint; //added - rpm 3/18/09

// Pull up file from central directory, if exists.

// Allows user to be notified of upgrades, etc.

// Get Home System Data for Restricted Pt List

If (EssoLoginYet <> True) And (ESSOVersion = true) And (listRestrictedPatients.count = 0) Then

Begin

EssoLoginYet := True;

frmSplashScreen.StatusLabel.Caption := 'Checking for restrictions...';

frmSplashScreen.Repaint;

TempStrServer := Uppercase(RPCBroker1.ANUStrServer);

TempStrPort := RPCBroker1.ANUStrPort;

RPCBroker1.Connected := False;

RPCBroker1.Server := RPCBroker1.ANUStrServerHome;

RPCBroker1.ListenerPort := StrtoInt(RPCBroker1.ANUStrPortHome);

//RPCBroker1.Connected := True; BSE mod - rpm 1/8/09

//If Not AuthorizedOption('DVBA CAPRI GUI') then begin BSE mod - rpm 1/8/09

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

//Populate Tools

FMListBoxCAPRITools.GetList;

If FMListBoxCAPRITools.Items.Count > 0 Then

Begin

For xx := 0 To FMListBoxCAPRITools.Items.Count - 1 Do

Begin

TempAction[xx] := TAction.Create(ActionManMain);

TempAction[xx].OnExecute := frmMain.tempToolClick;

TempAction[xx].Caption := Piece(FMListBoxCAPRITools.Items[xx], ';', 2);

TempAction[xx].Name := 'NewTool' + inttostr(xx);

TempAction[xx].Category := 'External Applications';

//Add weblink URL, if appropriate

If (pos('HTTP://', uppercase(FMListBoxCAPRITools.Items[xx])) > 0) or

(pos('HTTPS://', uppercase(FMListBoxCAPRITools.Items[xx])) > 0) Then

TempAction[xx].ImageIndex := 15

else

TempAction[xx].ImageIndex := -1;

tempAction[xx].ActionList := ActionManMain;

with ActionManMain.ActionBars[0].Items[2].Items[6].Items.Add do

Action := tempAction[xx];

End;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 5;

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

UserDUZHomeServer := RPCBroker1.Results[0];

//Get Security Keys

UserKeys.Clear;

FMListerUserKeys.IENS := ',' + UserDUZHomeServer;

FMListerUserKeys.GetList(UserKeys);

FMGets1.IENS := UserDUZHomeServer;

FMGets1.GetData;

homeemailaddress := lowercase(fmgets1.GetField('.151').FMDBExternal);

//Make sure .151 in the new person file has an entry

If Homeemailaddress = '' Then

Begin

{Init Mail}

{Get User E-Mail Address on Home Server}

RPCBroker1.RemoteProcedure := 'DVBAB MAIL INIT';

RPCBrokerCall;

Try

RPCBroker1.Call;

homeemailaddress := lowercase(Piece(RPCBroker1.Results[0], '^', 1));

frmForwardingAddress := TfrmForwardingAddress.Create(frmMain);

frmForwardingAddress.Edit1.text := 'NONE SPECIFIED';

frmForwardingAddress.RadioButton3.Visible := False;

frmForwardingAddress.RadioButton1.Caption := lowercase(homeemailaddress);

frmForwardingAddress.ShowModal;

If frmForwardingAddress.RadioButton1.Checked = True Then

Begin

homeemailaddress := lowercase(frmForwardingAddress.RadioButton1.Caption);

End;

If frmForwardingAddress.RadioButton2.Checked = True Then

Begin

homeemailaddress := lowercase(frmForwardingAddress.editOtherEmailAddress.text);

// Use specified e-mail address

End;

// Update .151 field next

FMFiler2.AddFDA('200', UserDUZHomeServer, '.151', HomeEMailAddress);

If Not fmfiler2.Update Then

Begin

fmfiler2.DisplayErrors;

End;

frmForwardingAddress.Release;

frmForwardingAddress := Nil;

Except On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Mail Init not completed.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

End;

End;

{Get User FilemanCode}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(0)'], nil, True) then

exit;

If RPCBroker1.Results.Count > 0 Then

UserFilemanCode := RPCBroker1.Results[0]

Else

UserFilemanCode := '';

//Check for restricted

ListRestrictedPatients.Clear;

{Get Division}

RPCBroker1.RemoteProcedure := 'DVBAB DIVISION';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DIVISION could not be accessed!');

End;

End;

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('You have not been given a division. Please contact IRM for assistance.');

application.terminate;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 10;

UserDivision := RPCBroker1.Results[0];

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ("2")'], nil, True) then

exit;

RemoteUserDivisionNumber := '';

FMGetsInstitution.IENS := RPCBroker1.Results[0];

// Check to be sure division doesn't come back as "1".

// If it does, read local field 999000, which is an override value

// for division.

FMGetsCAPRIDivision.IENS := UserDUZHomeServer;

FMGetsCAPRIDivision.GetData;

tempflag := false;

Try

If fmgetsCAPRIDivision.GetField('999000').FMDBExternal <> '' Then

tempflag := true Except tempflag := false;

End;

If tempflag = true Then

Begin

UserDivision := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 2);

UserDivisionNumber := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 1);

RemoteUserDivisionNumber := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 1);

End

Else

Begin

// Convert RPCBroker results to field 99 of institution file

FMGetsInstitution.GetData;

UserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

RemoteUserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

End;

If (RemoteUserDivisionNumber = '') Or (RemoteUserDivisionNumber = '1') Then

Begin

Showmessage('There was a problem detecting your division. Please notify IT support.');

application.terminate;

End;

// Strip "-" off end

TempString := '';

If Length(UserDivision) > 0 Then

For x := length(userdivision) Downto 1 Do

If Userdivision[x] = '-' Then

If (tempstring = '') Or (pos('-BVA/VBA', tempstring) > 0) Then

tempstring := Copy(Userdivision, 1, x - 1);

If TempString <> '' Then

UserDivision := TempString;

// --HERE--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

FMGetsNewPerson.IENS := RPCBroker1.Results[0];

frmSPlashScreen.GaugeLoadSite.Progress := 15;

FMGetsNewPerson.GetAndFill;

UserHomePrimaryMenu := FMGetsNewPerson.GetField('201').FMDBExternal;

FPrimaryMenuAssignable := IsPrimaryMenuAssignable(UserHomePrimaryMenu); //CodeCR109 - rpm 8/4/10

RestrictedPtSelection := FMGetsNewPerson.GetField('101.01').FMDBExternal;

PatientSelectionList := FMGetsNewPerson.GetField('101.02').FMDBInternal;

If RestrictedPtSelection = '' Then

RestrictedPtSelection := 'N';

If (RestrictedPtSelection[1] = 'Y') And (Uppercase(PatientSelectionList) = '') Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patient selection list.');

application.terminate;

End;

// 1/11/06

RestrictedPtSelection := 'N';

If RestrictedPtSelection[1] = 'Y' Then

Begin

frmSplashScreen.StatusLabel.Caption := 'Getting restricted patient list...';

frmSPlashScreen.GaugeLoadSite.Progress := 18;

frmSplashScreen.Repaint;

{Get User DUZ on Home System}

RPCBroker1.RemoteProcedure := 'DVBAB TEAM PATIENTS';

RPCBroker1.Param[0].Value := PatientSelectionList;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB TEAM PATIENTS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB TEAM PATIENTS could not be accessed!');

End;

End;

QuickCopy(frmMain.RPCBroker1.Results, listRestrictedPatients);

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('Your restricted patient list is empty.');

application.terminate;

End;

If Uppercase(listRestrictedPatients[0]) = '^NO PATIENTS FOUND.' Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patients on your list.');

application.terminate;

End;

End;

RPCBroker1.Connected := False;

RPCBroker1.Server := TempStrServer;

RPCBroker1.ListenerPort := StrtoInt(TempStrPort);

//RPCBroker1.Connected := True; BSE mod - rpm 1/8/09

//If Not AuthorizedOption('DVBA CAPRI GUI') Then BSE mod - rpm 1/8/09

if not ConnectToServer('DVBA CAPRI GUI') then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

End;

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

AuthorIEN := '';

If RPCBroker1.Results.Count > 0 Then

AuthorIEN := RPCBroker1.Results[0];

If ESSOVersion = False Then

Begin

//Get Security Keys

UserKeys.Clear;

If AuthorIEN <> '' Then

FMListerUserKeys.IENS := ',' + AuthorIEN;

FMListerUserKeys.GetList(UserKeys);

End;

UserKeysStr := UserKeys.Text;

FMGets1.IENS := AuthorIEN;

FMGets1.GetAndFill;

localemailaddress := fmgets1.GetField('.151').FMDBExternal;

AuthorSSN := FMGets1.GetField('9').FMDBInternal;

AuthorName := FMGets1.GetField('.01').FMDBExternal;

If ESSOVersion = False Then

Begin

{Get Division}

RPCBroker1.RemoteProcedure := 'DVBAB DIVISION';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DIVISION could not be accessed!');

End;

End;

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('You have not been given a division. Please contact IRM for assistance.');

application.terminate;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 22;

UserDivision := RPCBroker1.Results[0];

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(2)'], nil, True) then

exit;

// Convert RPCBroker results to field 99 of institution file

FMGetsInstitution.IENS := RPCBroker1.Results[0];

FMGetsInstitution.GetData;

UserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

// Strip "-" off end

TempString := '';

If Length(UserDivision) > 0 Then

For x := length(userdivision) Downto 1 Do

If Userdivision[x] = '-' Then

If (tempstring = '') Or (pos('-BVA/VBA', tempstring) > 0) Then

tempstring := Copy(Userdivision, 1, x - 1);

If TempString <> '' Then

UserDivision := TempString;

{Get User FilemanCode}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(0)'], nil, True) then

exit;

If RPCBroker1.Results.Count > 0 Then

UserFilemanCode := RPCBroker1.Results[0]

Else

UserFilemanCode := '';

end;

{telnet stuff -----------------------------------------------------}

// Used only in CAPRI Remote

// 201 = Primary Menu

// 7 = DISUSER - User not allowed to log on

// 9.2 = Termination date

UserRemoteDISUSER := '';

UserRemoteTerminationDate := '';

UserRemotePrimaryMenu := '';

If essoversion = true Then

Begin

FMGetsNewPerson.IENS := AuthorIEN;

FMGetsNewPerson.GetData;

// If error reading this, let the user in the system anyway

UserRemoteDISUSER := FMGetsNewPerson.GetField('7').FMDBInternal;

UserRemoteTerminationDate := FMGetsNewPerson.GetField('9.2').FMDBInternal;

UserRemotePrimaryMenu := FMGetsNewPerson.GetField('201').FMDBExternal;

// Check is user is HIA User.

// The essoselector controls populating the FMGetsComponent

try //protect against missing field #4 in file #396.96 - rpm 10/22/08

if formESSOSelect.FMGetsUserSites.GetField('4').FMDBInternal = 'Y' Then

UserRemoteHIAUser := True

else

UserRemoteHIAUser := False;

if formEssoSelect.FMGetsUserSites.GetField('5').FMDBInternal = 'Y' then

FIsDoDUser := True

else

FIsDoDUser := False;

except

//do nothing

end;

End;

{------------------------------------------------------------------}

frmSplashScreen.StatusLabel.Caption := 'Checking version...';

// frmSPlashScreen.GaugeLoadSite.Progress := 25;

frmSPlashScreen.GaugeLoadSite.Progress := 20;

frmSplashScreen.Repaint;

RPCBroker1.RemoteProcedure := 'DVBAB VERSION';

RPCBroker1.Param[0].Value := VersionUser;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB VERSION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB VERSION could not be accessed!');

application.terminate;

End;

End;

If RPCBroker1.Results.Count > 0 Then

Begin

// buffer := RPCBroker1.Results[0];

// ShowMessage(buffer);

VistAVersion := Piece(RPCBroker1.Results[0], '^', 1);

VersionA := StrToFloat(Piece(VistAVersion, '\*', 2));

VersionB := StrToFloat(Piece(VistAVersion, '\*', 3));

VersionRevisionSt := Piece(VistAVersion, '\*', 4);

XVersionA := StrToFloat(Piece(Version, '\*', 2));

XVersionB := StrToFloat(Piece(Version, '\*', 3));

XVersionRevisionSt := Piece(Version, '\*', 4);

if Uppercase(Piece(RPCBroker1.Results[0], '^', 2)) = 'NOOLD' then

IsOldVersionAllowed := False

else

IsOldVersionAllowed := True;

If Piece(RPCBroker1.Results[0], '^', 3) = 'SURGERY' Then

AllowSurgeryReport := True

Else

AllowSurgeryReport := False;

// Go ahead and hardcode this

AllowSurgeryReport := True;

// Make 45\*1 compatible with 41\*1 too

// 12/5/02 - Turn off compatibility. Require newest patch.

//ShowMessageCAPRI(floattostr(xversiona)+' '+floattostr(xversionb));

If FloatToStr(VersionA) = '41' Then

If FloatToStr(VersionB) = '1' Then

XVersionA := 41;

If XVersionA < VersionA Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if Assigned(frmSplashScreen) then

frmSplashScreen.Repaint;

if UserRemoteHIAUser = True then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

VersionShown := True;

End;

If XVersionA = VersionA Then

If XVersionB < VersionB Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if UserRemoteHIAUser = True Then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

VersionShown := True;

End;

If (VersionShown = False) And (XVersionRevisionSt < VersionRevisionSt) Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if UserRemoteHIAUser = True Then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

End;

End

Else

Begin

If VersionShown = False Then

ShowMessageCAPRI('The version loaded in VistA is corrupt or old. Please contact IRM for assistance. You may continue using CAPRI, but there could be problems while logged into this system.');

End;

// Restricted patient list for non-remote version

If EssoVersion = False Then

Begin

FMGetsNewPerson.IENS := AuthorIEN;

FMGetsNewPerson.GetAndFill;

RestrictedPtSelection := FMGetsNewPerson.GetField('101.01').FMDBExternal;

PatientSelectionList := FMGetsNewPerson.GetField('101.02').FMDBInternal;

If RestrictedPtSelection = '' Then

RestrictedPtSelection := 'N';

If (RestrictedPtSelection[1] = 'Y') And (Uppercase(PatientSelectionList) = '') Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patient selection list.');

application.terminate;

End;

If RestrictedPtSelection[1] = 'Y' Then

Begin

frmSplashScreen.StatusLabel.Caption := 'Getting restricted patient list...';

frmSPlashScreen.GaugeLoadSite.Progress := 30;

frmSplashScreen.Repaint;

RPCBroker1.RemoteProcedure := 'DVBAB TEAM PATIENTS';

RPCBroker1.Param[0].Value := PatientSelectionList;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB TEAM PATIENTS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB TEAM PATIENTS could not be accessed!');

End;

End;

If RPCBroker1.Results.Count > 0 Then

For xx := 0 To RPCBroker1.Results.Count - 1 Do

listRestrictedPatients.Add(RPCBroker1.Results[xx]);

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('Your restricted patient list is empty.');

application.terminate;

End;

If Uppercase(listRestrictedPatients[0]) = '^NO PATIENTS FOUND.' Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patients on your list.');

application.terminate;

End;

End;

If Not AuthorizedOption('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

End;

BitBtnNewForm.Visible := False;

// QC #2016 - jcs - Ensure button is always enabled to match right click delete.

// Delete event will validate.

// ButtonIPRDelete.Visible := False;

// QC #2016 - jcs - Variable no longer needed

// ANUCanEditCPWM := False;

// CodeCR668 - JRL 2/23/15

// It was discovered that there may be times when NO security keys are

// assigned and that allows some menu options to display that shouldn't.

// To avoid code and testing impact, I'm modifying the test from:

// If UserKeys.Count > 0 Then

// to

// If UserKeys.Count >= 0 Then

// This mimizes code changes late in the testing cycle. The test will

// always succeed and all code in the loop below will always be executed.

If UserKeys.Count >= 0 Then

Begin

ReadOnlyMode := IsUserKeyInList('DVBA CAPRI READ ONLY,DVBA CAPRI READ O');

actToolsExamListParameters.Visible := IsUserKeyInList('DVBA CAPRI EXAM LIST EDIT,DVBA CAPRI EXAM LIST E,DVBA C SUPERVISOR');

If IsUserKeyInList('DVBA CAPRI WORKSHEET TAB,DVBA CAPRI WORKSHEET ') Then

Begin

TabCPWorksheets.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

actToolsUnsignedWorksheets.Visible := True;

End;

// Check Patch Number for defensive coding

if IsPatchInstalled('DVBA\*2.7\*181') then // CodeCR347 - JRL 6/12/12

begin // CodeCR347 - JRL 6/12/12

// Make the vocational rehab tab visible only if the user // CodeCR347 - JRL 6/12/12

// has VocRehab userkeys // CodeCR347 - JRL 6/12/12

TabVocRehab.TabVisible := IsUserKeyInList('DVBA CAPRI VRE\_COUNSELOR,DVBA CAPRI VRE\_COUNSELOR'); // CodeCR347 - JRL 6/12/12

if TabVocRehab.TabVisible then // CodeCR347 - JRL 6/12/12

begin // CodeCR347 - JRL 6/12/12

if not Assigned(VocRehab) then // CodeCR347 - JRL 6/12/12

VocRehab := TVocRehab.Create; // CodeCR347 - JRL 6/12/12

// assign access if user has DHN So Rehab privileges // CodeCR347 - JRL 6/12/12

// (used for assigning consults) // CodeCR347 - JRL 6/12/12

VocRehab.VHAAccess := IsUserKeyInList('DVBA CAPRI VHA\_COORDINATOR,DVBA CAPRI VHA\_COORDINATOR'); // CodeCR347 - JRL 6/12/12

end; // CodeCR347 - JRL 6/12/12

end // CodeCR347 - JRL 6/12/12

else // CodeCR347 - JRL 6/12/12

TabVocRehab.TabVisible := FALSE; // CodeCR347 - JRL 6/12/12

// DVBA CAPRI DENY GETVBADOCS was added in patch 186. VBA users could

// be assigned this key. If this key was assigned, then the menu option

// 'Get Docs from VVA' would not be visible. After implementation, this

// was determined to be hard to administrate, so this key is being

// deprecated and a new key will be added in patch 187 to all users

// to allow them access to 'Get Docs from VVA'. VBA users not allowed

// to see VVA docs here will not get this key. -JRL 5/15/14

if IsPatchInstalled('DVBA\*2.7\*186') AND // CodeCR577 - JRL 5/15/14 //Only execute this code if

NOT IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR577 - JRL 5/15/14 //Patch 186 is installed and

begin // CodeCR577 - JRL 5/15/14 //Patch 187 is not installed

if IsUserKeyInList('DVBA CAPRI DENY\_GETVBADOCS') then // CodeCR497 - JRL 6/07/13

actFileRetrieveVirtualVA.Visible := FALSE // CodeCR497 - JRL 6/07/13

else // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := TRUE; // CodeCR577 - JRL 5/15/14

end; // CodeCR577 - JRL 5/15/14

if IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR577 - JRL 5/15/14 //Only execute this code if

begin // CodeCR577 - JRL 5/15/14 //if Patch 187 is installed

if IsUserKeyInList('DVBA CAPRI GETVBADOCS') then // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := TRUE // CodeCR577 - JRL 5/15/14

else // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := FALSE; // CodeCR577 - JRL 5/15/14

end; // CodeCR577 - JRL 5/15/14

// if IsUserKeyInList('DVBA GetDocsfromVLER') then // CodeCR567 - JRL 5/18/14

// actFileRetrieveDocsVLERDAS.Visible := TRUE // CodeCR567 - JRL 5/18/14

// else // CodeCR567 - JRL 5/18/14

// actFileRetrieveDocsVLERDAS.Visible := FALSE; // CodeCR567 - JRL 5/18/14

// Add in security key for retrieving documents from VLER/DAS // CodeCR567 - JRL 5/22/14

// You must have this key to see the menu option Get Docs from VLER // CodeCR567 - JRL 5/22/14

if IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR567 - JRL 5/22/14

begin // CodeCR567 - JRL 5/22/14

// get the separator between Get Docs from Virtual VA and Get Docs from DAS

try

actnItemSeparator := ActionManMain.ActionBars[0].Items[0].Items[11]; // CodeCR668 - RPK 1/21/2015

except

actnItemSeparator := nil;

end;

if IsUserKeyInList('DVBA CAPRI GETDOCSFROMVLER') then // CodeCR567 - JRL 5/18/14

begin // CodeCR567 - JRL 5/22/14

actFileRetrieveDocsVLERDAS.Visible := TRUE; // CodeCR567 - JRL 5/18/14

// actFileSeparatorBarVisible.Visible := TRUE; // CodeCR567 - JRL 5/22/14

if actnItemSeparator <> nil then // CodeCR668 - RPK 1/21/2015

actnItemSeparator.Visible := True; // CodeCR668 - RPK 1/21/2015

end // CodeCR567 - JRL 5/22/14

else // CodeCR567 - JRL 5/18/14

begin // CodeCR567 - JRL 5/22/14

actFileRetrieveDocsVLERDAS.Visible := FALSE; // CodeCR567 - JRL 5/18/14

// actFileSeparatorBarVisible.Visible := FALSE; // CodeCR567 - JRL 5/22/14

if actnItemSeparator <> nil then // CodeCR668 - RPK 1/21/2015

actnItemSeparator.Visible := False; // CodeCR668 - RPK 1/21/2015

end // CodeCR567 - JRL 5/22/14

end; // CodeCR567 - JRL 5/22/14

if IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR698 - RPK 3/30/2015

begin // CodeCR698 - RPK 3/30/2015

if ESSOVersion then // If remotely logged in, then check security key // CodeCR698 - JRL 4/20/2015

begin // CodeCR698 - JRL 4/20/2015

actHelpEdtCancelReasons.Visible := IsUserKey('DVBA CAPRI EXAMCANC'); // CodeCR698 - RPK 3/30/2015

actHelpEdtInsuffReasons.Visible := IsUserKey('DVBA CAPRI EXAMINSUFF'); // CodeCR698 - RPK 3/30/2015

end // CodeCR698 - JRL 4/20/2015

else // CodeCR698 - JRL 4/20/2015

begin // CodeCR698 - JRL 4/20/2015

actHelpEdtCancelReasons.Visible := FALSE; // CodeCR698 - JRL 4/20/2015

actHelpEdtInsuffReasons.Visible := FALSE; // CodeCR698 - JRL 4/20/2015

end; // CodeCR698 - JRL 4/20/2015

end; // CodeCR698 - RPK 3/30/2015

FoundCPWMKey := '';

If IsUserKeyInList('DVBAB CPWM REQUIRE REVIEW,DVBAB CPWM REQUIRE REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM REQUIRE REVIEW,';

If IsUserKeyInList('DVBAB CPWM DISALLOW REVIEW,DVBAB CPWM DISALLOW REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM DISALLOW REVIEW,';

If IsUserKeyInList('DVBAB CPWM OPTIONAL REVIEW,DVBAB CPWM OPTIONAL REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM OPTIONAL REVIEW,';

boReviewerKey := IsUserKeyInList('DVBAB CPWM REVIEWER,DVBAB CPWM REVIE');

If (FoundCPWMKey <> '') Or boReviewerKey Then

Begin

BitBtnNewForm.Visible := True;

// QC #2016 - jcs - Ensure button is always enabled to match right click delete.

// Delete event will validate.

//ButtonIPRDelete.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

TabCPWorksheets.Visible := True;

actToolsCPRSCosigUtil.Visible := True;

actToolsUnsignedWorksheets.Visible := True;

End;

If Piece(FoundCPWMKey, ',', 2) <> '' Then

Begin

ShowmessageCAPRI('You have conflicting CAPRI security keys. Please notify IRM there is a problem with keys: ' + char(13) + Copy(FoundCPWMKey, 1, length(FoundCPWMKey) - 1));

End;

End;

// Enable C&P Worksheets tab for CLAIMS users with FileMan access "@", "U", and "u".

if (TabCPWorksheets.TabVisible = False) and (EssoVersion) then

begin

if (Pos('@', userfilemancode) > 0) or

(Pos('u', userfilemancode) > 0) or

(Pos('U', userfilemancode) > 0) then

begin

TabCPWorksheets.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

end;

end;

//Loading Lab tests used to be here...

ListLabTestNames.Clear;

frmSplashScreen.StatusLabel.Caption := 'Loading drug classes...';

fmlistboxvadrugclass.getlist;

If fmlistervadrugclass.more = true Then

Repeat fmlistboxvadrugclass.getmore

Until fmlistervadrugclass.more = false;

frmSplashScreen.StatusLabel.Caption := 'Loading medical center divisions...';

frmSPlashScreen.GaugeLoadSite.Progress := 65;

frmSplashScreen.Repaint;

// Get divisions and force them in the order they are in the file

GetAllDivisions(ListMedicalCenterDivision); //CodeCR186 - rpm 4/18/11

frmSplashScreen.StatusLabel.Caption := 'Loading exam list...';

frmSPlashScreen.GaugeLoadSite.Progress := 75;

frmSplashScreen.Repaint;

FMListerExams.GetList(listExams); // Get Exams

//Load Institutions Used to be here...

lstInstitutions.Items.Clear;

frmROFinder.ORListBoxSites.Items.Clear;

frmSPlashScreen.GaugeLoadSite.Progress := 100;

frmSplashScreen.Repaint;

{Load lab tests}

//frmLabGraph.FMListBoxLabs.GetList;

//timerLoadLabs.Enabled:=True;

If ESSOVersion = False Then

Begin

{Get DTime}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DTIME'], nil, True) then

exit;

TimeOutVal := StrToInt(RPCBroker1.Results[0]);

If TimeOutVal < 600 Then

TimeOutVal := 600;

End;

TimeOutTimer.Enabled := True;

//If Pos('@',userfilemancode)>0 then EditelectronicSignatureCode1.Visible:=True;

If Pos('@', userfilemancode) > 0 Then

Begin

actToolsExamListParameters.Visible := True;

actToolsCPRSCosigUtil.Visible := True;

//TabCPWorksheets.Visible:=True;

//TabCPWorksheets.Enabled:=True;

//TabCPWorksheets.TabVisible:=True;

//UnsignedCPWorksheets1.Visible:=True;

End;

//Check for mail

If frmMailMan = Nil Then

frmMailMan := TfrmMailMan.Create(frmMain);

frmMailMan.StringGridMailboxMessages.RowCount := 0;

frmMailMan.StringGridMailboxMessages.Cells[0, 0] := ''; // Clear these in case no messages are in the mailbox

frmMailMan.StringGridMailboxMessages.Cells[1, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[2, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[3, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[4, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[5, 0] := ''; // Not used for now

frmMailMan.StringGridMailboxMessages.Cells[6, 0] := '';

frmMailMan.btnRefreshClick(Application);

//Now check to see if they have a forwarding e-mail address in

//file 200. If not, prompt them.

If ESSOVersion = True Then

If lowercase(localemailaddress) <> lowercase(homeemailaddress) Then

Begin

FMFiler2.AddFDA('200', AuthorIEN, '.151', HomeEMailAddress);

LocalEMailAddress := lowercase(homeemailaddress);

Try

If Not fmfiler2.Update Then

Begin

fmfiler2.DisplayErrors;

End;

ShowMessageCAPRI('Remote e-mail address for this system has been' + char(13) + 'updated to match your home server preference:' + char(13) + HomeEmailAddress);

Except

Begin

ShowmessageCAPRI('Couldn''t set remote e-mail address.');

If RPCBroker1.Connected = False Then

Begin

ShowmessageCAPRI('Not connected to a server due to data transmission failure. Exiting application.');

application.Terminate;

application.terminate;

End;

End;

End;

End;

frmSPlashScreen.GaugeLoadSite.ForeColor := clYellow;

frmAlerts := TfrmAlerts.Create(frmMain);

if IsCCRUser then begin

// frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Manifests available for Pending / Under Review...';

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Vendor exam requests ready for download...'; // CodeCR730 rpk 6/2/2015

frmSPlashScreen.GaugeLoadSite.Progress := 20;

frmSplashScreen.Repaint;

frmAlerts.btnManifests.Show;

frmAlerts.edtManifests.Show;

frmAlerts.shpManifests.Show; // rpk 11/5/2014

frmManifests := TfrmManifests.Create(frmMain);

try

ManifestCnt := frmManifests.getManifestCnt;

// edtstr := 'Unable to retrieve vendor exam requests pending review.';

edtstr := 'Unable to retrieve vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

if ManifestCnt >= 0 then begin

case ManifestCnt of

// 0: edtstr := 'You have no vendor exam requests pending review.';

// 1: edtstr := 'You have 1 vendor exam requests pending review.';

0: edtstr := 'You have no vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

1: edtstr := 'You have 1 vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

else

// edtstr := 'You have ' + IntToStr(ManifestCnt) + ' vendor exam requests pending review.';

edtstr := 'You have ' + IntToStr(ManifestCnt) + ' vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

end;

end;

frmAlerts.edtManifests.Text := edtstr;

frmAlerts.btnManifests.Enabled := ManifestCnt > 0;

frmAlerts.edtManifests.Enabled := ManifestCnt > 0;

finally

FreeAndNil(frmManifests); // rpk 7/11/2014

end;

end

else begin

frmAlerts.btnManifests.Hide;

frmAlerts.edtManifests.Hide;

frmAlerts.shpManifests.Hide; // rpk 11/5/2014

end;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: CPRS co-signature status...';

frmSPlashScreen.GaugeLoadSite.Progress := 40;

frmSplashScreen.Repaint;

// Check for documents ready to be transferred from TIU to AMIE

If actToolsCPRSCosigUtil.visible = true Then

Begin

frmAlerts.buttonResolve.Show; // rpk 11/6/2014

frmAlerts.editResolve.Show; // rpk 11/6/2014

// frmAlerts.shape4.Visible := false;

frmAlerts.shpResolve.Show; // rpk 11/6/2014

frmUncosigned := TfrmUncosigned.Create(frmMain);

frmUncosigned.buttonSearchClick(Application);

If piece(frmUncosigned.labelReady.caption, ' ', 1) = '0' Then

Begin

frmAlerts.editResolve.Text := 'There are no CPRS co-signed documents ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := False;

frmAlerts.ButtonResolve.Enabled := False;

End

Else

If piece(frmUncosigned.labelReady.caption, ' ', 1) = '1' Then

Begin

frmAlerts.editResolve.Text := 'There is 1 CPRS co-signed document ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := True;

frmAlerts.ButtonResolve.Enabled := True;

End

Else

Begin

frmAlerts.editResolve.Text := 'There are ' + piece(frmUncosigned.labelReady.caption, ' ', 1) + ' CPRS co-signed documents ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := True;

frmAlerts.ButtonResolve.Enabled := True;

End;

End

Else

Begin

// Hide this one

frmAlerts.buttonResolve.Visible := false;

frmAlerts.editResolve.Visible := false;

// frmAlerts.shape4.Visible := false;

frmAlerts.shpResolve.Visible := false;

// frmAlerts.BitBtnContinue.top := 162;

// frmAlerts.Height := 240;

// have stopped moving Continue button and resizing Alerts form; just leave Resolve row hidden; rpk 11/6/2014

// frmAlerts.BitBtnContinue.top := frmAlerts.BitBtnContinue.top - 50; // rpk 11/5/2014

// frmAlerts.Height := frmAlerts.Height - 44;

//frmAlerts.height:=frmAlerts.bitBtnContinue.Top+frmAlerts.bitBtnContinue.Height+16;

End;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Unsigned templates...';

// frmSPlashScreen.GaugeLoadSite.Progress := 50;

frmSPlashScreen.GaugeLoadSite.Progress := 60;

frmSplashScreen.Repaint;

// Check for forms needing signature

timeouttimer.enabled := true;

UnsignedView := TUnsignedView.Create(frmMain);

UnsignedView.Button5Click(Application);

If ((UnsignedView.FMListBoxIPR1.Items.Count > 0) And (ReadOnlyMode = False)) Then

Begin

// Duplicate check for documents needing signature, same as in alerts

frmAlerts.ButtonUnsignedClick(frmMain);

End;

// Check for forms needing signature if user has access to the worksheet tab

For xx := 0 To UserKeys.Count - 1 Do

If (UserKeys[xx] = 'DVBA CAPRI WORKSHEET TAB') Or

(UserKeys[xx] = 'DVBA CAPRI WORKSHEET ') Then

Begin

Flag := False;

//For xx:=0 to UserKeys.Count-1 do ShowMessageCAPRI(userkeys[xx]);

If UserKeys.Count > 0 Then

Begin

//For xx:=0 to UserKeys.Count-1 do ShowMessageCAPRI(UserKeys[xx]);

For yy := 0 To UserKeys.Count - 1 Do

If (UserKeys[yy] = 'DVBAB CPWM REVIEWER') Or

(UserKeys[yy] = 'DVBAB CPWM REVIE') Then

Begin

flag := true;

End;

End;

If Flag = True Then

Begin

frmSplashScreen.Show;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Templates ready for review...';

// frmSPlashScreen.GaugeLoadSite.Progress := 75;

frmSPlashScreen.GaugeLoadSite.Progress := 80;

frmSplashScreen.Repaint;

UnsignedView.Button4Click(frmMain);

If ((UnsignedView.FMListBoxIPR1.Items.Count > 0) And (ReadOnlyMode = False)) Then

Begin

If UnsignedView.FMListBoxIPR1.Items.Count = 1 Then

frmAlerts.editReview.Text := 'There is 1 template requiring review.'

Else

Begin

zz := 0;

For x := 0 To UnsignedView.FMListBoxIPR1.Items.Count - 1 Do

If Copy(UnsignedView.FMListBoxIPR1.Items[x], 1, 1) <> '!' Then

inc(zz);

If zz > 1 Then

frmAlerts.editReview.Text := 'There are ' + inttostr(zz) + ' templates requiring review.';

If zz = 1 Then

frmAlerts.EditReview.Text := 'There is 1 template requiring review.';

End;

frmAlerts.buttonReview.enabled := True;

frmAlerts.editReview.enabled := True;

End

Else

Begin

frmAlerts.editReview.enabled := false;

frmAlerts.editReview.text := 'There are no templates requiring review.';

frmAlerts.buttonReview.enabled := False;

End;

End

Else

Begin

frmAlerts.editReview.enabled := false;

frmAlerts.editReview.text := 'You do not have template "reviewer" privileges.';

frmAlerts.buttonReview.enabled := False;

End;

End;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: CPRS un-cosigned documents...';

frmSPlashScreen.GaugeLoadSite.Progress := 100;

frmSplashScreen.Repaint;

frmTIUCosign := TfrmTIUCosign.Create(Nil);

frmTIUCosign.GetCosignatureAlerts;

End;

If ScreenReaderActive = True Then

frmMain.Caption := 'CAPRI'

Else

frmMain.Caption := ProgramNameCaption + ' Connected To ' + Uppercase(RPCBroker1.ANUStrServer) + ' (Server:' + RPCBroker1.ANURPCServer + ' Volume:' + RPCBroker1.ANURPCVolume + ' UCI:' + RPCBroker1.ANURPCUCI + ' Port:' + RPCBroker1.ANURPCPort +

')';

labeldivision.caption := '';

If ESSOVersion = True Then

Begin

labeldivision.visible := true;

RPCBroker1.RemoteProcedure := 'DVBAB SET DIVISION';

RPCBroker1.Param[1].Value := RemoteUserDivisionNumber;

RPCBroker1.Param[1].PType := literal;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB SET DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB SET DIVISION could not be accessed!');

End;

End;

Try

If piece(rpcbroker1.results[0], '^', 1) = '0' Then

Begin

LabelDivision.Caption := 'Division: Set division failed!';

ShowmessageCAPRI('WARNING: There was a problem setting the user''s division. Some reports may not function correctly. ' + piece(rpcbroker1.results[0], '^', 2));

End

Else

Begin

// User's division IEN is in piece 1, but we don't need it right now.

LabelDivision.caption := 'Division: ' + piece(rpcbroker1.results[0], '^', 2);

End;

Except

Begin

End;

End;

End

Else

Begin

// Non-remote mode

labelDivision.Caption := 'Division: ' + UserDivision;

End;

PanelDivision.Width := labelDivision.Width + 8;

If labelDivision.Caption = '' Then

Begin

ShowmessageCAPRI('WARNING: There was a problem setting the user''s division. Please check the host system and log in again. If the problem continues, notify your IT support service.');

timerhalt.enabled := true;

End;

Panel1.Visible := True;

SwitchToSite := ''; //Clear this, just in case it was previously set

AnimateLogo(False);

If RPCBroker1.Connected = False Then

Begin

frmMain.Caption := 'CAPRI not Connected'

End;

End

Else

Begin

TimeOutVal := 0;

TimeOutTimer.Enabled := False;

RPCBroker1.Connected := False;

ClearForm; //CodeCR248 - rpm 3/8/12 - clear the header and tab set

Panel1.Visible := False;

AnimateLogo(False);

End;

If Assigned(frmSplashScreen) Then

Begin

frmSplashScreen.Hide;

frmSplashScreen.release;

frmSplashScreen := Nil;

End;

If Assigned(frmAlerts) Then

Begin

If (frmAlerts.buttonUnsigned.enabled = true) Or (frmAlerts.buttonReview.enabled = true)

Or (frmAlerts.buttonUncosigned.enabled = true) Or (frmAlerts.buttonResolve.enabled = true)

Or (frmAlerts.btnManifests.enabled = true) // 2014-11-03 LMS Fix bug to show alert dialog when manifests are available

Then

Begin

frmAlerts.ShowModal;

End;

frmAlerts.Hide;

frmAlerts.release;

frmAlerts := Nil;

End;

If frmMain.Visible Then

//SelectPatient1Click(Sender);

actFileSelectPatientExecute(Sender);

For x := 1 To 100 Do

application.processmessages;

If Assigned(frmTIUCosign) Then

Begin

FreeAndNil(frmTIUCosign);

End;

If Assigned(frmUncosigned) Then

Begin

frmUncosigned.Hide;

Try

frmUncosigned.release;

Except

End;

frmUncosigned := Nil;

End;

If UnsignedView <> Nil Then

Begin

Try

UnsignedView.Release;

Except

End;

UnsignedView := Nil;

End;

switchtosite := '';

ContextorChangeMessage := '';

CCOWBreakLink := False;

// Check if CPEP has been notified that this used has used this version before

// Only do it one time.

tempstring2 := version;

If pos('\*', tempstring2) > 0 Then

Repeat

tempstring2 := copy(tempstring2, 1, pos('\*', tempstring2) - 1) + '\_' + copy(tempstring2, pos('\*', tempstring2) + 1, 255);

Until pos('\*', tempstring2) = 0;

If Not fileexists(tempdir + tempstring2 + '.validated') Then

Begin

tempstring := '';

tempstring := authorname + '^' + UserDivision + '^' + UserDivisionNumber + '^' + version;

aStream := TMemoryStream.Create;

formNews.GetURL('http://DNS cpepa/cgi-bin/capri\_track.pl?' + tempstring, aStream);

formNews.RichEditNews.Lines.LoadFromStream(aStream);

formNews.RichEditNews.Lines.SaveToFile(tempdir + tempstring2 + '.validated');

formNews.RichEditNews.Lines.Clear;

formNews.GetURL('http://' + newsURL + 'NewsHeaders.txt', aStream);

formNews.RichEditNews.Lines.LoadFromStream(aStream);

aStream := nil;

End;

End;

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

procedure TfrmMain.actFileConnectExecute(Sender: TObject);

Var

boReviewerKey: Boolean;

Flag: Boolean;

FoundCPWMKey: String;

IsOldVersionAllowed: Boolean;

ManifestCnt: Integer; // rpk 7/2/2014

tempflag: boolean;

TempPort: Integer;

TempSender: TAction;

TempServer: String;

tempstring: String;

TempString2: String;

TempStrPort: String;

TempStrServer: String;

versiona: real;

versionb: real;

versionrevisionst: String;

versionshown: boolean;

VistAVersion: String;

x: integer;

xversiona: real;

xversionb: real;

xversionrevisionst: String;

xx: integer;

yy: integer;

zz: integer;

tempClientItem: TActionClientItem;

aStream: TMemoryStream;

authToken: string;

VDUrl: string;

edtstr: String;

actnItemSeparator: TActionClientItem; // CodeCR668 rpk 1/21/2015

Begin

// Code "1892890Dhdoy\*(@hDH28y(#\*yUHKJDH"

If ANURemoteProcedureCallInProgress = True Then

exit;

If CCOWMode = False Then

Try

If RPCBroker1.Contextor <> Nil Then

RPCBroker1.Contextor := Nil;

Except

End;

If Essoversion = false Then

labelDivision.visible := False;

If Assigned(formTIUDisplay) Then

Begin

Try

formTIUDisplay.Visible := False;

formTIUDisplay.Release;

formTIUDisplay := Nil;

Except

End;

End;

If (actFileConnect.caption = '&Disconnect...') And

Assigned(PNCSForm) Then

Begin

ShowMessage('Please close the open Template.');

PNCSForm.Show;

Exit;

End;

//AnimateLogo(True);

TempSender := Nil;

If Uppercase(Sender.Classname) = 'TACTION' Then

Begin

TempSender := Sender As TAction;

End;

versionshown := false;

If (actFileConnect.Caption = '&Connect...') Or ((Assigned(TempSender)) And (TempSender = actFileSwitchSites)) Then

Begin //connect

// Dec 2005 Chuck Sener Reports that the GUI sometimes hangs at this spot and never

// shuts down

// Put in a 60 second timer and close CAPRI if no connection in that time.

TimerInitialConnection.Enabled := True;

If SwitchToSite = '' Then

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShuttingDown := True;

timerhalt.enabled := true;

application.terminate;

exit;

End;

// If it gets to this spot, the connection worked, so turn off the timer.

TimerInitialConnection.Enabled := False;

If SwitchToSite = '' Then

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

//timerhalt.enabled:=true;

application.terminate;

exit;

End;

If formESSOSelect = Nil Then

Begin

formESSOSelect := TformESSOSelect.Create(frmMain);

HomeServer := RPCBroker1.Server;

HomePort := InttoStr(RPCBroker1.ListenerPort);

//CodeCR353 - rpm 5/14/12

//Populate Virtual VA URL value on initial connection to home server

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['6'], nil) then

begin

if RPCBroker1.Results.Count > 0 then

begin

FVirtualVAURL := RPCBroker1.Results[0];

//CodeCR353 - rpm 5/31/12

//A URL result indicates that we can populate the Virtual VA TOKEN

//value because the server patch is loaded at the site.

if (FVirtualVAURL <> '') then

begin

if CallRPC(RPCBroker1, 'DVBAB GET VVA TOKEN', [], nil) then

if RPCBroker1.Results.Count > 0 then

FVirtualVAToken := RPCBroker1.Results[0];

end;

end;

end;

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['7'], nil) then // CodeCR440 (AMIS290) JRL 11/15/12

begin // CodeCR440 (AMIS290) JRL 11/15/12

if RPCBroker1.Results.Count > 0 then // CodeCR440 (AMIS290) JRL 11/15/12

begin // CodeCR440 (AMIS290) JRL 11/15/12

FAMIS290URL := RPCBroker1.Results[0]; // CodeCR440 (AMIS290) JRL 11/15/12

//If an URL exists, this indicates that the AMIS290 // CodeCR440 (AMIS290) JRL 11/15/12

//reports have been replaced. New AMIS290 functionality // CodeCR440 (AMIS290) JRL 11/15/12

//is found at the URL location // CodeCR440 (AMIS290) JRL 11/15/12

end; // CodeCR440 (AMIS290) JRL 11/15/12

end; // CodeCR440 (AMIS290) JRL 11/15/12

if TIdHTTPVler.GetVlerDasURL() = '' then //could have been set via command line param //CodeCR500 JRL 9/10/13

begin //CodeCR500 JRL 9/10/13

if CallRPC(RPCBroker1, 'DVBAB GET URL', ['8'], nil) then //CodeCR500 JRL 08/27/13

begin //CodeCR500 JRL 08/27/13

if RPCBroker1.Results.Count > 0 then //CodeCR500 JRL 08/27/13

begin //CodeCR500 JRL 08/27/13

TIdHTTPVler.SetVlerDasURL(RPCBroker1.Results[0]); //CodeCR??? LMS 2014-07-21 Moving from main form field. Needs to be available App wide in all units that need VLER access

end; //CodeCR500 JRL 08/27/13

end; //CodeCR500 JRL 08/27/13

end;

End;

TempServer := RPCBroker1.Server;

TempPort := RPCBroker1.ListenerPort;

SetFont;

If frmMailMessage = Nil Then

frmMailMessage := TfrmMailMessage.Create(frmMain);

If ESSOVersion = True Then

Begin

TimeOutTimer.Enabled := True;

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DTIME'], nil, True) then

exit;

If TimeOutVal < 0 Then // Need this to keep value from getting reset when switching sites

TimeOutVal := StrToInt(RPCBroker1.Results[0]);

If TimeOutVal < 600 Then

TimeOutVal := 600;

If frmMailMan = Nil Then

Begin

// Only check mail on home server once at start-up

frmMailMan := TfrmMailMan.Create(frmMain);

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010 - Why are we making this call?

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

If frmMailMan = Nil Then

frmMailMan := TfrmMailMan.Create(frmMain);

setfont;

End;

setfont;

formEssoSelect.StaticText1.Caption := 'Select a site.';

ContextorChangeMessage := 'The application is in the process of switching VistA accounts. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

FormEssoSelect.formstyle := fsStayOnTop;

If FormEssoSelect.Showmodal = mrcancel Then

Begin // Check for cancel button and exit

If ESSOInitialized = True Then

If RPCBroker1.Connected = True Then

exit;

ShuttingDown := True;

ContextorChangeMessage := '';

CCOWBreakLink := False;

actFileQuitExecute(Application);

End;

// CodeCR185 -MER 04/2011

if Assigned(ReflectionWrapper) then

ReflectionWrapper.TerminateEmulator;

// Drop telnet connection if one exists

If assigned(frmTelnet) Then

If frmTelnet.bitbtnDisconnect.Visible = True Then

Begin

frmTelnet.DisconnectBtnClick(Self);

End;

ESSOConnecting := True;

ESSOInitialized := True;

If (TempServer = RPCBroker1.Server) And (TempPort = RPCBroker1.ListenerPort) Then

Begin

exit;

End;

TimerRemoteData.Enabled := False;

ListBoxRemoteDataPending.Items.Clear;

If frmMain.RPCBroker1.Server = '' Then

Begin

application.terminate;

End;

tempClientItem := ActionManMain.FindItemByCaption('Ex&ternal Applications');

If Assigned(tempClientItem) Then

tempClientItem.Visible := True;

//Set HIA Search menu Item under Tools menu to enable true

// MNJ - 09/10/2012

actHIASearch.Enabled := True;

End

else

begin

tempClientItem := ActionManMain.FindItemByCaption('Ex&ternal Applications');

If Assigned(tempClientItem) Then

tempClientItem.Visible := False;

end;

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

ESSOConnecting := False;

If RPCBroker1.Connected = True Then

Begin

TimeOutTimer.Enabled := True;

PatientIEN := '';

patientIENDOD := '';

frmMain.Caption := ProgramNameCaption;

ClearForm; //CodeCR248 - rpm 3/8/12 - clear the header and tab set

If CCOWMode = True Then

Begin

//If ESSOVersion<>True then begin

// Get local station number

RPCBroker1.RemoteProcedure := 'DVBAB CCOW';

RPCBroker1.Param[0].Value := '1';

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'DVBAB CCOW could not be accessed!';

ProgressBarLoadPt.Repaint;

frmMain.Repaint;

ShowMessageCAPRI('DVBAB CCOW could not be accessed!');

application.terminate;

application.terminate;

End;

End;

LocalStationNumber := Piece(RPCBroker1.Results[0], '^', 3);

// Get Test system status

RPCBroker1.RemoteProcedure := 'DVBAB CCOW';

RPCBroker1.Param[0].Value := '2';

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'DVBAB CCOW could not be accessed!';

ProgressBarLoadPt.Repaint;

frmMain.Repaint;

ShowMessageCAPRI('DVBAB CCOW could not be accessed!');

application.terminate;

application.terminate;

End;

End;

TestSystemLogin := RPCBroker1.Results[0];

//End;

End;

{Reload Divisions and Institutions on New System}

frmSPlashScreen := TfrmSplashScreen.Create(frmMain);

frmSplashScreen.PanelVersion.Height := Panel1.Height;

frmSplashScreen.PanelVersion.Caption := VersionUser;

frmSplashScreen.PanelVersion.Left := (frmSPlashScreen.Width Div 2) - (frmSPlashScreen.PanelVersion.Width Div 2);

frmSPlashScreen.StatusLabel.Height := Panel1.Height;

frmSPlashScreen.Shape1.Height := Panel1.Height;

frmSPlashScreen.GaugeLoadSite.Progress := 0;

frmSPlashScreen.GaugeLoadSite.ForeColor := clRed;

frmSPlashScreen.GaugeLoadSite.Top := frmSPlashScreen.Shape1.Top + frmSPlashScreen.Shape1.Height;

frmSPlashScreen.Height := frmSPlashScreen.GaugeLoadSite.Top + 32;

frmSPlashScreen.Panel1.Height := frmSPlashScreen.GaugeLoadSite.Top + 33;

frmSPlashScreen.StatusLabel.Font := panel1.font;

frmSplashScreen.Show;

frmSplashScreen.Repaint; //added - rpm 3/18/09

// Pull up file from central directory, if exists.

// Allows user to be notified of upgrades, etc.

// Get Home System Data for Restricted Pt List

If (EssoLoginYet <> True) And (ESSOVersion = true) And (listRestrictedPatients.count = 0) Then

Begin

EssoLoginYet := True;

frmSplashScreen.StatusLabel.Caption := 'Checking for restrictions...';

frmSplashScreen.Repaint;

TempStrServer := Uppercase(RPCBroker1.ANUStrServer);

TempStrPort := RPCBroker1.ANUStrPort;

RPCBroker1.Connected := False;

RPCBroker1.Server := RPCBroker1.ANUStrServerHome;

RPCBroker1.ListenerPort := StrtoInt(RPCBroker1.ANUStrPortHome);

//RPCBroker1.Connected := True; BSE mod - rpm 1/8/09

//If Not AuthorizedOption('DVBA CAPRI GUI') then begin BSE mod - rpm 1/8/09

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

//Populate Tools

FMListBoxCAPRITools.GetList;

If FMListBoxCAPRITools.Items.Count > 0 Then

Begin

For xx := 0 To FMListBoxCAPRITools.Items.Count - 1 Do

Begin

TempAction[xx] := TAction.Create(ActionManMain);

TempAction[xx].OnExecute := frmMain.tempToolClick;

TempAction[xx].Caption := Piece(FMListBoxCAPRITools.Items[xx], ';', 2);

TempAction[xx].Name := 'NewTool' + inttostr(xx);

TempAction[xx].Category := 'External Applications';

//Add weblink URL, if appropriate

If (pos('HTTP://', uppercase(FMListBoxCAPRITools.Items[xx])) > 0) or

(pos('HTTPS://', uppercase(FMListBoxCAPRITools.Items[xx])) > 0) Then

TempAction[xx].ImageIndex := 15

else

TempAction[xx].ImageIndex := -1;

tempAction[xx].ActionList := ActionManMain;

with ActionManMain.ActionBars[0].Items[2].Items[6].Items.Add do

Action := tempAction[xx];

End;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 5;

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

UserDUZHomeServer := RPCBroker1.Results[0];

//Get Security Keys

UserKeys.Clear;

FMListerUserKeys.IENS := ',' + UserDUZHomeServer;

FMListerUserKeys.GetList(UserKeys);

FMGets1.IENS := UserDUZHomeServer;

FMGets1.GetData;

homeemailaddress := lowercase(fmgets1.GetField('.151').FMDBExternal);

//Make sure .151 in the new person file has an entry

If Homeemailaddress = '' Then

Begin

{Init Mail}

{Get User E-Mail Address on Home Server}

RPCBroker1.RemoteProcedure := 'DVBAB MAIL INIT';

RPCBrokerCall;

Try

RPCBroker1.Call;

homeemailaddress := lowercase(Piece(RPCBroker1.Results[0], '^', 1));

frmForwardingAddress := TfrmForwardingAddress.Create(frmMain);

frmForwardingAddress.Edit1.text := 'NONE SPECIFIED';

frmForwardingAddress.RadioButton3.Visible := False;

frmForwardingAddress.RadioButton1.Caption := lowercase(homeemailaddress);

frmForwardingAddress.ShowModal;

If frmForwardingAddress.RadioButton1.Checked = True Then

Begin

homeemailaddress := lowercase(frmForwardingAddress.RadioButton1.Caption);

End;

If frmForwardingAddress.RadioButton2.Checked = True Then

Begin

homeemailaddress := lowercase(frmForwardingAddress.editOtherEmailAddress.text);

// Use specified e-mail address

End;

// Update .151 field next

FMFiler2.AddFDA('200', UserDUZHomeServer, '.151', HomeEMailAddress);

If Not fmfiler2.Update Then

Begin

fmfiler2.DisplayErrors;

End;

frmForwardingAddress.Release;

frmForwardingAddress := Nil;

Except On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Mail Init not completed.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

End;

End;

{Get User FilemanCode}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(0)'], nil, True) then

exit;

If RPCBroker1.Results.Count > 0 Then

UserFilemanCode := RPCBroker1.Results[0]

Else

UserFilemanCode := '';

//Check for restricted

ListRestrictedPatients.Clear;

{Get Division}

RPCBroker1.RemoteProcedure := 'DVBAB DIVISION';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DIVISION could not be accessed!');

End;

End;

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('You have not been given a division. Please contact IRM for assistance.');

application.terminate;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 10;

UserDivision := RPCBroker1.Results[0];

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ("2")'], nil, True) then

exit;

RemoteUserDivisionNumber := '';

FMGetsInstitution.IENS := RPCBroker1.Results[0];

// Check to be sure division doesn't come back as "1".

// If it does, read local field 999000, which is an override value

// for division.

FMGetsCAPRIDivision.IENS := UserDUZHomeServer;

FMGetsCAPRIDivision.GetData;

tempflag := false;

Try

If fmgetsCAPRIDivision.GetField('999000').FMDBExternal <> '' Then

tempflag := true Except tempflag := false;

End;

If tempflag = true Then

Begin

UserDivision := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 2);

UserDivisionNumber := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 1);

RemoteUserDivisionNumber := Piece(uppercase(fmgetsCAPRIDivision.GetField('999000').FMDBExternal), ';', 1);

End

Else

Begin

// Convert RPCBroker results to field 99 of institution file

FMGetsInstitution.GetData;

UserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

RemoteUserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

End;

If (RemoteUserDivisionNumber = '') Or (RemoteUserDivisionNumber = '1') Then

Begin

Showmessage('There was a problem detecting your division. Please notify IT support.');

application.terminate;

End;

// Strip "-" off end

TempString := '';

If Length(UserDivision) > 0 Then

For x := length(userdivision) Downto 1 Do

If Userdivision[x] = '-' Then

If (tempstring = '') Or (pos('-BVA/VBA', tempstring) > 0) Then

tempstring := Copy(Userdivision, 1, x - 1);

If TempString <> '' Then

UserDivision := TempString;

// --HERE--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

FMGetsNewPerson.IENS := RPCBroker1.Results[0];

frmSPlashScreen.GaugeLoadSite.Progress := 15;

FMGetsNewPerson.GetAndFill;

UserHomePrimaryMenu := FMGetsNewPerson.GetField('201').FMDBExternal;

FPrimaryMenuAssignable := IsPrimaryMenuAssignable(UserHomePrimaryMenu); //CodeCR109 - rpm 8/4/10

RestrictedPtSelection := FMGetsNewPerson.GetField('101.01').FMDBExternal;

PatientSelectionList := FMGetsNewPerson.GetField('101.02').FMDBInternal;

If RestrictedPtSelection = '' Then

RestrictedPtSelection := 'N';

If (RestrictedPtSelection[1] = 'Y') And (Uppercase(PatientSelectionList) = '') Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patient selection list.');

application.terminate;

End;

// 1/11/06

RestrictedPtSelection := 'N';

If RestrictedPtSelection[1] = 'Y' Then

Begin

frmSplashScreen.StatusLabel.Caption := 'Getting restricted patient list...';

frmSPlashScreen.GaugeLoadSite.Progress := 18;

frmSplashScreen.Repaint;

{Get User DUZ on Home System}

RPCBroker1.RemoteProcedure := 'DVBAB TEAM PATIENTS';

RPCBroker1.Param[0].Value := PatientSelectionList;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB TEAM PATIENTS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB TEAM PATIENTS could not be accessed!');

End;

End;

QuickCopy(frmMain.RPCBroker1.Results, listRestrictedPatients);

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('Your restricted patient list is empty.');

application.terminate;

End;

If Uppercase(listRestrictedPatients[0]) = '^NO PATIENTS FOUND.' Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patients on your list.');

application.terminate;

End;

End;

RPCBroker1.Connected := False;

RPCBroker1.Server := TempStrServer;

RPCBroker1.ListenerPort := StrtoInt(TempStrPort);

//RPCBroker1.Connected := True; BSE mod - rpm 1/8/09

//If Not AuthorizedOption('DVBA CAPRI GUI') Then BSE mod - rpm 1/8/09

if not ConnectToServer('DVBA CAPRI GUI') then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

End;

{Get User DUZ}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ'], nil, True) then

exit;

AuthorIEN := '';

If RPCBroker1.Results.Count > 0 Then

AuthorIEN := RPCBroker1.Results[0];

If ESSOVersion = False Then

Begin

//Get Security Keys

UserKeys.Clear;

If AuthorIEN <> '' Then

FMListerUserKeys.IENS := ',' + AuthorIEN;

FMListerUserKeys.GetList(UserKeys);

End;

UserKeysStr := UserKeys.Text;

FMGets1.IENS := AuthorIEN;

FMGets1.GetAndFill;

localemailaddress := fmgets1.GetField('.151').FMDBExternal;

AuthorSSN := FMGets1.GetField('9').FMDBInternal;

AuthorName := FMGets1.GetField('.01').FMDBExternal;

If ESSOVersion = False Then

Begin

{Get Division}

RPCBroker1.RemoteProcedure := 'DVBAB DIVISION';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DIVISION could not be accessed!');

End;

End;

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('You have not been given a division. Please contact IRM for assistance.');

application.terminate;

End;

frmSPlashScreen.GaugeLoadSite.Progress := 22;

UserDivision := RPCBroker1.Results[0];

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(2)'], nil, True) then

exit;

// Convert RPCBroker results to field 99 of institution file

FMGetsInstitution.IENS := RPCBroker1.Results[0];

FMGetsInstitution.GetData;

UserDivisionNumber := FMGetsInstitution.GetField('99').FMDBExternal;

// Strip "-" off end

TempString := '';

If Length(UserDivision) > 0 Then

For x := length(userdivision) Downto 1 Do

If Userdivision[x] = '-' Then

If (tempstring = '') Or (pos('-BVA/VBA', tempstring) > 0) Then

tempstring := Copy(Userdivision, 1, x - 1);

If TempString <> '' Then

UserDivision := TempString;

{Get User FilemanCode}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DUZ(0)'], nil, True) then

exit;

If RPCBroker1.Results.Count > 0 Then

UserFilemanCode := RPCBroker1.Results[0]

Else

UserFilemanCode := '';

end;

{telnet stuff -----------------------------------------------------}

// Used only in CAPRI Remote

// 201 = Primary Menu

// 7 = DISUSER - User not allowed to log on

// 9.2 = Termination date

UserRemoteDISUSER := '';

UserRemoteTerminationDate := '';

UserRemotePrimaryMenu := '';

If essoversion = true Then

Begin

FMGetsNewPerson.IENS := AuthorIEN;

FMGetsNewPerson.GetData;

// If error reading this, let the user in the system anyway

UserRemoteDISUSER := FMGetsNewPerson.GetField('7').FMDBInternal;

UserRemoteTerminationDate := FMGetsNewPerson.GetField('9.2').FMDBInternal;

UserRemotePrimaryMenu := FMGetsNewPerson.GetField('201').FMDBExternal;

// Check is user is HIA User.

// The essoselector controls populating the FMGetsComponent

try //protect against missing field #4 in file #396.96 - rpm 10/22/08

if formESSOSelect.FMGetsUserSites.GetField('4').FMDBInternal = 'Y' Then

UserRemoteHIAUser := True

else

UserRemoteHIAUser := False;

if formEssoSelect.FMGetsUserSites.GetField('5').FMDBInternal = 'Y' then

FIsDoDUser := True

else

FIsDoDUser := False;

except

//do nothing

end;

End;

{------------------------------------------------------------------}

frmSplashScreen.StatusLabel.Caption := 'Checking version...';

// frmSPlashScreen.GaugeLoadSite.Progress := 25;

frmSPlashScreen.GaugeLoadSite.Progress := 20;

frmSplashScreen.Repaint;

RPCBroker1.RemoteProcedure := 'DVBAB VERSION';

RPCBroker1.Param[0].Value := VersionUser;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB VERSION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB VERSION could not be accessed!');

application.terminate;

End;

End;

If RPCBroker1.Results.Count > 0 Then

Begin

// buffer := RPCBroker1.Results[0];

// ShowMessage(buffer);

VistAVersion := Piece(RPCBroker1.Results[0], '^', 1);

VersionA := StrToFloat(Piece(VistAVersion, '\*', 2));

VersionB := StrToFloat(Piece(VistAVersion, '\*', 3));

VersionRevisionSt := Piece(VistAVersion, '\*', 4);

XVersionA := StrToFloat(Piece(Version, '\*', 2));

XVersionB := StrToFloat(Piece(Version, '\*', 3));

XVersionRevisionSt := Piece(Version, '\*', 4);

if Uppercase(Piece(RPCBroker1.Results[0], '^', 2)) = 'NOOLD' then

IsOldVersionAllowed := False

else

IsOldVersionAllowed := True;

If Piece(RPCBroker1.Results[0], '^', 3) = 'SURGERY' Then

AllowSurgeryReport := True

Else

AllowSurgeryReport := False;

// Go ahead and hardcode this

AllowSurgeryReport := True;

// Make 45\*1 compatible with 41\*1 too

// 12/5/02 - Turn off compatibility. Require newest patch.

//ShowMessageCAPRI(floattostr(xversiona)+' '+floattostr(xversionb));

If FloatToStr(VersionA) = '41' Then

If FloatToStr(VersionB) = '1' Then

XVersionA := 41;

If XVersionA < VersionA Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if Assigned(frmSplashScreen) then

frmSplashScreen.Repaint;

if UserRemoteHIAUser = True then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

VersionShown := True;

End;

If XVersionA = VersionA Then

If XVersionB < VersionB Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if UserRemoteHIAUser = True Then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

VersionShown := True;

End;

If (VersionShown = False) And (XVersionRevisionSt < VersionRevisionSt) Then

Begin

If VersionShown = False Then

ShowMessageCAPRI('Your client software is older than the server on VistA. Please contact IRM for assistance. VistA is running ' + VistAVersion);

if NOT IsOldVersionAllowed then

begin

//ShowMessageCAPRI('Old versions of CAPRI will not run from this server.');

ShowMessageCAPRI(

'Versions of CAPRI prior to DVBA\*2.7\*' + FloatToStr(VersionA) + ' will not run from this server.' + #13 + #10 +

'You are currently running DVBA\*2.7\*' + FloatToStr(XVersionA) + '.');

if UserRemoteHIAUser = True Then

ShowHIAUserDownloadWebsite;

application.terminate;

end;

End;

End

Else

Begin

If VersionShown = False Then

ShowMessageCAPRI('The version loaded in VistA is corrupt or old. Please contact IRM for assistance. You may continue using CAPRI, but there could be problems while logged into this system.');

End;

// Restricted patient list for non-remote version

If EssoVersion = False Then

Begin

FMGetsNewPerson.IENS := AuthorIEN;

FMGetsNewPerson.GetAndFill;

RestrictedPtSelection := FMGetsNewPerson.GetField('101.01').FMDBExternal;

PatientSelectionList := FMGetsNewPerson.GetField('101.02').FMDBInternal;

If RestrictedPtSelection = '' Then

RestrictedPtSelection := 'N';

If (RestrictedPtSelection[1] = 'Y') And (Uppercase(PatientSelectionList) = '') Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patient selection list.');

application.terminate;

End;

If RestrictedPtSelection[1] = 'Y' Then

Begin

frmSplashScreen.StatusLabel.Caption := 'Getting restricted patient list...';

frmSPlashScreen.GaugeLoadSite.Progress := 30;

frmSplashScreen.Repaint;

RPCBroker1.RemoteProcedure := 'DVBAB TEAM PATIENTS';

RPCBroker1.Param[0].Value := PatientSelectionList;

RPCBroker1.Param[0].PType := literal;

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB TEAM PATIENTS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB TEAM PATIENTS could not be accessed!');

End;

End;

If RPCBroker1.Results.Count > 0 Then

For xx := 0 To RPCBroker1.Results.Count - 1 Do

listRestrictedPatients.Add(RPCBroker1.Results[xx]);

If RPCBroker1.Results.Count = 0 Then

Begin

ShowMessageCAPRI('Your restricted patient list is empty.');

application.terminate;

End;

If Uppercase(listRestrictedPatients[0]) = '^NO PATIENTS FOUND.' Then

Begin

ShowMessageCAPRI('You are restricted to a set of patients but have no patients on your list.');

application.terminate;

End;

End;

If Not AuthorizedOption('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

End;

BitBtnNewForm.Visible := False;

// QC #2016 - jcs - Ensure button is always enabled to match right click delete.

// Delete event will validate.

// ButtonIPRDelete.Visible := False;

// QC #2016 - jcs - Variable no longer needed

// ANUCanEditCPWM := False;

// CodeCR668 - JRL 2/23/15

// It was discovered that there may be times when NO security keys are

// assigned and that allows some menu options to display that shouldn't.

// To avoid code and testing impact, I'm modifying the test from:

// If UserKeys.Count > 0 Then

// to

// If UserKeys.Count >= 0 Then

// This mimizes code changes late in the testing cycle. The test will

// always succeed and all code in the loop below will always be executed.

If UserKeys.Count >= 0 Then

Begin

ReadOnlyMode := IsUserKeyInList('DVBA CAPRI READ ONLY,DVBA CAPRI READ O');

actToolsExamListParameters.Visible := IsUserKeyInList('DVBA CAPRI EXAM LIST EDIT,DVBA CAPRI EXAM LIST E,DVBA C SUPERVISOR');

If IsUserKeyInList('DVBA CAPRI WORKSHEET TAB,DVBA CAPRI WORKSHEET ') Then

Begin

TabCPWorksheets.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

actToolsUnsignedWorksheets.Visible := True;

End;

// Check Patch Number for defensive coding

if IsPatchInstalled('DVBA\*2.7\*181') then // CodeCR347 - JRL 6/12/12

begin // CodeCR347 - JRL 6/12/12

// Make the vocational rehab tab visible only if the user // CodeCR347 - JRL 6/12/12

// has VocRehab userkeys // CodeCR347 - JRL 6/12/12

TabVocRehab.TabVisible := IsUserKeyInList('DVBA CAPRI VRE\_COUNSELOR,DVBA CAPRI VRE\_COUNSELOR'); // CodeCR347 - JRL 6/12/12

if TabVocRehab.TabVisible then // CodeCR347 - JRL 6/12/12

begin // CodeCR347 - JRL 6/12/12

if not Assigned(VocRehab) then // CodeCR347 - JRL 6/12/12

VocRehab := TVocRehab.Create; // CodeCR347 - JRL 6/12/12

// assign access if user has DHN So Rehab privileges // CodeCR347 - JRL 6/12/12

// (used for assigning consults) // CodeCR347 - JRL 6/12/12

VocRehab.VHAAccess := IsUserKeyInList('DVBA CAPRI VHA\_COORDINATOR,DVBA CAPRI VHA\_COORDINATOR'); // CodeCR347 - JRL 6/12/12

end; // CodeCR347 - JRL 6/12/12

end // CodeCR347 - JRL 6/12/12

else // CodeCR347 - JRL 6/12/12

TabVocRehab.TabVisible := FALSE; // CodeCR347 - JRL 6/12/12

// DVBA CAPRI DENY GETVBADOCS was added in patch 186. VBA users could

// be assigned this key. If this key was assigned, then the menu option

// 'Get Docs from VVA' would not be visible. After implementation, this

// was determined to be hard to administrate, so this key is being

// deprecated and a new key will be added in patch 187 to all users

// to allow them access to 'Get Docs from VVA'. VBA users not allowed

// to see VVA docs here will not get this key. -JRL 5/15/14

if IsPatchInstalled('DVBA\*2.7\*186') AND // CodeCR577 - JRL 5/15/14 //Only execute this code if

NOT IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR577 - JRL 5/15/14 //Patch 186 is installed and

begin // CodeCR577 - JRL 5/15/14 //Patch 187 is not installed

if IsUserKeyInList('DVBA CAPRI DENY\_GETVBADOCS') then // CodeCR497 - JRL 6/07/13

actFileRetrieveVirtualVA.Visible := FALSE // CodeCR497 - JRL 6/07/13

else // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := TRUE; // CodeCR577 - JRL 5/15/14

end; // CodeCR577 - JRL 5/15/14

if IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR577 - JRL 5/15/14 //Only execute this code if

begin // CodeCR577 - JRL 5/15/14 //if Patch 187 is installed

if IsUserKeyInList('DVBA CAPRI GETVBADOCS') then // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := TRUE // CodeCR577 - JRL 5/15/14

else // CodeCR577 - JRL 5/15/14

actFileRetrieveVirtualVA.Visible := FALSE; // CodeCR577 - JRL 5/15/14

end; // CodeCR577 - JRL 5/15/14

// if IsUserKeyInList('DVBA GetDocsfromVLER') then // CodeCR567 - JRL 5/18/14

// actFileRetrieveDocsVLERDAS.Visible := TRUE // CodeCR567 - JRL 5/18/14

// else // CodeCR567 - JRL 5/18/14

// actFileRetrieveDocsVLERDAS.Visible := FALSE; // CodeCR567 - JRL 5/18/14

// Add in security key for retrieving documents from VLER/DAS // CodeCR567 - JRL 5/22/14

// You must have this key to see the menu option Get Docs from VLER // CodeCR567 - JRL 5/22/14

if IsPatchInstalled('DVBA\*2.7\*187') then // CodeCR567 - JRL 5/22/14

begin // CodeCR567 - JRL 5/22/14

// get the separator between Get Docs from Virtual VA and Get Docs from DAS

try

actnItemSeparator := ActionManMain.ActionBars[0].Items[0].Items[11]; // CodeCR668 - RPK 1/21/2015

except

actnItemSeparator := nil;

end;

if IsUserKeyInList('DVBA CAPRI GETDOCSFROMVLER') then // CodeCR567 - JRL 5/18/14

begin // CodeCR567 - JRL 5/22/14

actFileRetrieveDocsVLERDAS.Visible := TRUE; // CodeCR567 - JRL 5/18/14

// actFileSeparatorBarVisible.Visible := TRUE; // CodeCR567 - JRL 5/22/14

if actnItemSeparator <> nil then // CodeCR668 - RPK 1/21/2015

actnItemSeparator.Visible := True; // CodeCR668 - RPK 1/21/2015

end // CodeCR567 - JRL 5/22/14

else // CodeCR567 - JRL 5/18/14

begin // CodeCR567 - JRL 5/22/14

actFileRetrieveDocsVLERDAS.Visible := FALSE; // CodeCR567 - JRL 5/18/14

// actFileSeparatorBarVisible.Visible := FALSE; // CodeCR567 - JRL 5/22/14

if actnItemSeparator <> nil then // CodeCR668 - RPK 1/21/2015

actnItemSeparator.Visible := False; // CodeCR668 - RPK 1/21/2015

end // CodeCR567 - JRL 5/22/14

end; // CodeCR567 - JRL 5/22/14

if IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR698 - RPK 3/30/2015

begin // CodeCR698 - RPK 3/30/2015

if ESSOVersion then // If remotely logged in, then check security key // CodeCR698 - JRL 4/20/2015

begin // CodeCR698 - JRL 4/20/2015

actHelpEdtCancelReasons.Visible := IsUserKey('DVBA CAPRI EXAMCANC'); // CodeCR698 - RPK 3/30/2015

actHelpEdtInsuffReasons.Visible := IsUserKey('DVBA CAPRI EXAMINSUFF'); // CodeCR698 - RPK 3/30/2015

end // CodeCR698 - JRL 4/20/2015

else // CodeCR698 - JRL 4/20/2015

begin // CodeCR698 - JRL 4/20/2015

actHelpEdtCancelReasons.Visible := FALSE; // CodeCR698 - JRL 4/20/2015

actHelpEdtInsuffReasons.Visible := FALSE; // CodeCR698 - JRL 4/20/2015

end; // CodeCR698 - JRL 4/20/2015

end; // CodeCR698 - RPK 3/30/2015

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 JRL 7/29/16**

**begin // Patch 193 JRL 7/29/16**

**if (IsUserKey('sec')) AND // Patch 193 JRL 7/29/16**

**(ESSOVersion = FALSE) then // If remotely logged in, then check security key // CodeCR698 - JRL 4/20/2015**

**actHelpEdtReRouteReasons.Visible := TRUE // Patch 193 JRL 7/29/16**

**else // Patch 193 JRL 7/29/16**

**actHelpEdtReRouteReasons.Visible := FALSE; // Patch 193 JRL 7/29/16**

**end // Patch 193 JRL 7/29/16**

**else // Patch 193 JRL 9/23/16**

**actHelpEdtReRouteReasons.Visible := FALSE; // Patch 193 JRL 9/23/16**

FoundCPWMKey := '';

If IsUserKeyInList('DVBAB CPWM REQUIRE REVIEW,DVBAB CPWM REQUIRE REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM REQUIRE REVIEW,';

If IsUserKeyInList('DVBAB CPWM DISALLOW REVIEW,DVBAB CPWM DISALLOW REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM DISALLOW REVIEW,';

If IsUserKeyInList('DVBAB CPWM OPTIONAL REVIEW,DVBAB CPWM OPTIONAL REV') Then

FoundCPWMKey := FoundCPWMKey + 'DVBAB CPWM OPTIONAL REVIEW,';

boReviewerKey := IsUserKeyInList('DVBAB CPWM REVIEWER,DVBAB CPWM REVIE');

If (FoundCPWMKey <> '') Or boReviewerKey Then

Begin

BitBtnNewForm.Visible := True;

// QC #2016 - jcs - Ensure button is always enabled to match right click delete.

// Delete event will validate.

//ButtonIPRDelete.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

TabCPWorksheets.Visible := True;

actToolsCPRSCosigUtil.Visible := True;

actToolsUnsignedWorksheets.Visible := True;

End;

If Piece(FoundCPWMKey, ',', 2) <> '' Then

Begin

ShowmessageCAPRI('You have conflicting CAPRI security keys. Please notify IRM there is a problem with keys: ' + char(13) + Copy(FoundCPWMKey, 1, length(FoundCPWMKey) - 1));

End;

End;

// Enable C&P Worksheets tab for CLAIMS users with FileMan access "@", "U", and "u".

if (TabCPWorksheets.TabVisible = False) and (EssoVersion) then

begin

if (Pos('@', userfilemancode) > 0) or

(Pos('u', userfilemancode) > 0) or

(Pos('U', userfilemancode) > 0) then

begin

TabCPWorksheets.Visible := True;

TabCPWorksheets.Enabled := True;

TabCPWorksheets.TabVisible := True;

end;

end;

//Loading Lab tests used to be here...

ListLabTestNames.Clear;

frmSplashScreen.StatusLabel.Caption := 'Loading drug classes...';

fmlistboxvadrugclass.getlist;

If fmlistervadrugclass.more = true Then

Repeat fmlistboxvadrugclass.getmore

Until fmlistervadrugclass.more = false;

frmSplashScreen.StatusLabel.Caption := 'Loading medical center divisions...';

frmSPlashScreen.GaugeLoadSite.Progress := 65;

frmSplashScreen.Repaint;

// Get divisions and force them in the order they are in the file

GetAllDivisions(ListMedicalCenterDivision); //CodeCR186 - rpm 4/18/11

frmSplashScreen.StatusLabel.Caption := 'Loading exam list...';

frmSPlashScreen.GaugeLoadSite.Progress := 75;

frmSplashScreen.Repaint;

FMListerExams.GetList(listExams); // Get Exams

//Load Institutions Used to be here...

lstInstitutions.Items.Clear;

frmROFinder.ORListBoxSites.Items.Clear;

frmSPlashScreen.GaugeLoadSite.Progress := 100;

frmSplashScreen.Repaint;

{Load lab tests}

//frmLabGraph.FMListBoxLabs.GetList;

//timerLoadLabs.Enabled:=True;

If ESSOVersion = False Then

Begin

{Get DTime}

//CAPRI\_CodeCR95 - jcs - 05/20/2010

if not CallRPC(RPCBroker1, 'XWB GET VARIABLE VALUE', ['DTIME'], nil, True) then

exit;

TimeOutVal := StrToInt(RPCBroker1.Results[0]);

If TimeOutVal < 600 Then

TimeOutVal := 600;

End;

TimeOutTimer.Enabled := True;

//If Pos('@',userfilemancode)>0 then EditelectronicSignatureCode1.Visible:=True;

If Pos('@', userfilemancode) > 0 Then

Begin

actToolsExamListParameters.Visible := True;

actToolsCPRSCosigUtil.Visible := True;

//TabCPWorksheets.Visible:=True;

//TabCPWorksheets.Enabled:=True;

//TabCPWorksheets.TabVisible:=True;

//UnsignedCPWorksheets1.Visible:=True;

End;

//Check for mail

If frmMailMan = Nil Then

frmMailMan := TfrmMailMan.Create(frmMain);

frmMailMan.StringGridMailboxMessages.RowCount := 0;

frmMailMan.StringGridMailboxMessages.Cells[0, 0] := ''; // Clear these in case no messages are in the mailbox

frmMailMan.StringGridMailboxMessages.Cells[1, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[2, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[3, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[4, 0] := '';

frmMailMan.StringGridMailboxMessages.Cells[5, 0] := ''; // Not used for now

frmMailMan.StringGridMailboxMessages.Cells[6, 0] := '';

frmMailMan.btnRefreshClick(Application);

//Now check to see if they have a forwarding e-mail address in

//file 200. If not, prompt them.

If ESSOVersion = True Then

If lowercase(localemailaddress) <> lowercase(homeemailaddress) Then

Begin

FMFiler2.AddFDA('200', AuthorIEN, '.151', HomeEMailAddress);

LocalEMailAddress := lowercase(homeemailaddress);

Try

If Not fmfiler2.Update Then

Begin

fmfiler2.DisplayErrors;

End;

ShowMessageCAPRI('Remote e-mail address for this system has been' + char(13) + 'updated to match your home server preference:' + char(13) + HomeEmailAddress);

Except

Begin

ShowmessageCAPRI('Couldn''t set remote e-mail address.');

If RPCBroker1.Connected = False Then

Begin

ShowmessageCAPRI('Not connected to a server due to data transmission failure. Exiting application.');

application.Terminate;

application.terminate;

End;

End;

End;

End;

frmSPlashScreen.GaugeLoadSite.ForeColor := clYellow;

frmAlerts := TfrmAlerts.Create(frmMain);

if IsCCRUser then begin

// frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Manifests available for Pending / Under Review...';

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Vendor exam requests ready for download...'; // CodeCR730 rpk 6/2/2015

frmSPlashScreen.GaugeLoadSite.Progress := 20;

frmSplashScreen.Repaint;

frmAlerts.btnManifests.Show;

frmAlerts.edtManifests.Show;

frmAlerts.shpManifests.Show; // rpk 11/5/2014

frmManifests := TfrmManifests.Create(frmMain);

try

ManifestCnt := frmManifests.getManifestCnt;

// edtstr := 'Unable to retrieve vendor exam requests pending review.';

edtstr := 'Unable to retrieve vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

if ManifestCnt >= 0 then begin

case ManifestCnt of

// 0: edtstr := 'You have no vendor exam requests pending review.';

// 1: edtstr := 'You have 1 vendor exam requests pending review.';

0: edtstr := 'You have no vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

1: edtstr := 'You have 1 vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

else

// edtstr := 'You have ' + IntToStr(ManifestCnt) + ' vendor exam requests pending review.';

edtstr := 'You have ' + IntToStr(ManifestCnt) + ' vendor exam requests ready for download.'; // CodeCR730 rpk 6/2/2015

end;

end;

frmAlerts.edtManifests.Text := edtstr;

frmAlerts.btnManifests.Enabled := ManifestCnt > 0;

frmAlerts.edtManifests.Enabled := ManifestCnt > 0;

finally

FreeAndNil(frmManifests); // rpk 7/11/2014

end;

end

else begin

frmAlerts.btnManifests.Hide;

frmAlerts.edtManifests.Hide;

frmAlerts.shpManifests.Hide; // rpk 11/5/2014

end;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: CPRS co-signature status...';

frmSPlashScreen.GaugeLoadSite.Progress := 40;

frmSplashScreen.Repaint;

// Check for documents ready to be transferred from TIU to AMIE

If actToolsCPRSCosigUtil.visible = true Then

Begin

frmAlerts.buttonResolve.Show; // rpk 11/6/2014

frmAlerts.editResolve.Show; // rpk 11/6/2014

// frmAlerts.shape4.Visible := false;

frmAlerts.shpResolve.Show; // rpk 11/6/2014

frmUncosigned := TfrmUncosigned.Create(frmMain);

frmUncosigned.buttonSearchClick(Application);

If piece(frmUncosigned.labelReady.caption, ' ', 1) = '0' Then

Begin

frmAlerts.editResolve.Text := 'There are no CPRS co-signed documents ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := False;

frmAlerts.ButtonResolve.Enabled := False;

End

Else

If piece(frmUncosigned.labelReady.caption, ' ', 1) = '1' Then

Begin

frmAlerts.editResolve.Text := 'There is 1 CPRS co-signed document ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := True;

frmAlerts.ButtonResolve.Enabled := True;

End

Else

Begin

frmAlerts.editResolve.Text := 'There are ' + piece(frmUncosigned.labelReady.caption, ' ', 1) + ' CPRS co-signed documents ready for transfer to AMIE.';

frmAlerts.editResolve.Enabled := True;

frmAlerts.ButtonResolve.Enabled := True;

End;

End

Else

Begin

// Hide this one

frmAlerts.buttonResolve.Visible := false;

frmAlerts.editResolve.Visible := false;

// frmAlerts.shape4.Visible := false;

frmAlerts.shpResolve.Visible := false;

// frmAlerts.BitBtnContinue.top := 162;

// frmAlerts.Height := 240;

// have stopped moving Continue button and resizing Alerts form; just leave Resolve row hidden; rpk 11/6/2014

// frmAlerts.BitBtnContinue.top := frmAlerts.BitBtnContinue.top - 50; // rpk 11/5/2014

// frmAlerts.Height := frmAlerts.Height - 44;

//frmAlerts.height:=frmAlerts.bitBtnContinue.Top+frmAlerts.bitBtnContinue.Height+16;

End;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Unsigned templates...';

// frmSPlashScreen.GaugeLoadSite.Progress := 50;

frmSPlashScreen.GaugeLoadSite.Progress := 60;

frmSplashScreen.Repaint;

// Check for forms needing signature

timeouttimer.enabled := true;

UnsignedView := TUnsignedView.Create(frmMain);

UnsignedView.Button5Click(Application);

If ((UnsignedView.FMListBoxIPR1.Items.Count > 0) And (ReadOnlyMode = False)) Then

Begin

// Duplicate check for documents needing signature, same as in alerts

frmAlerts.ButtonUnsignedClick(frmMain);

End;

// Check for forms needing signature if user has access to the worksheet tab

For xx := 0 To UserKeys.Count - 1 Do

If (UserKeys[xx] = 'DVBA CAPRI WORKSHEET TAB') Or

(UserKeys[xx] = 'DVBA CAPRI WORKSHEET ') Then

Begin

Flag := False;

//For xx:=0 to UserKeys.Count-1 do ShowMessageCAPRI(userkeys[xx]);

If UserKeys.Count > 0 Then

Begin

//For xx:=0 to UserKeys.Count-1 do ShowMessageCAPRI(UserKeys[xx]);

For yy := 0 To UserKeys.Count - 1 Do

If (UserKeys[yy] = 'DVBAB CPWM REVIEWER') Or

(UserKeys[yy] = 'DVBAB CPWM REVIE') Then

Begin

flag := true;

End;

End;

If Flag = True Then

Begin

frmSplashScreen.Show;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: Templates ready for review...';

// frmSPlashScreen.GaugeLoadSite.Progress := 75;

frmSPlashScreen.GaugeLoadSite.Progress := 80;

frmSplashScreen.Repaint;

UnsignedView.Button4Click(frmMain);

If ((UnsignedView.FMListBoxIPR1.Items.Count > 0) And (ReadOnlyMode = False)) Then

Begin

If UnsignedView.FMListBoxIPR1.Items.Count = 1 Then

frmAlerts.editReview.Text := 'There is 1 template requiring review.'

Else

Begin

zz := 0;

For x := 0 To UnsignedView.FMListBoxIPR1.Items.Count - 1 Do

If Copy(UnsignedView.FMListBoxIPR1.Items[x], 1, 1) <> '!' Then

inc(zz);

If zz > 1 Then

frmAlerts.editReview.Text := 'There are ' + inttostr(zz) + ' templates requiring review.';

If zz = 1 Then

frmAlerts.EditReview.Text := 'There is 1 template requiring review.';

End;

frmAlerts.buttonReview.enabled := True;

frmAlerts.editReview.enabled := True;

End

Else

Begin

frmAlerts.editReview.enabled := false;

frmAlerts.editReview.text := 'There are no templates requiring review.';

frmAlerts.buttonReview.enabled := False;

End;

End

Else

Begin

frmAlerts.editReview.enabled := false;

frmAlerts.editReview.text := 'You do not have template "reviewer" privileges.';

frmAlerts.buttonReview.enabled := False;

End;

End;

frmSplashScreen.StatusLabel.Caption := 'Checking Alert: CPRS un-cosigned documents...';

frmSPlashScreen.GaugeLoadSite.Progress := 100;

frmSplashScreen.Repaint;

frmTIUCosign := TfrmTIUCosign.Create(Nil);

frmTIUCosign.GetCosignatureAlerts;

End;

If ScreenReaderActive = True Then

frmMain.Caption := 'CAPRI'

Else

frmMain.Caption := ProgramNameCaption + ' Connected To ' + Uppercase(RPCBroker1.ANUStrServer) + ' (Server:' + RPCBroker1.ANURPCServer + ' Volume:' + RPCBroker1.ANURPCVolume + ' UCI:' + RPCBroker1.ANURPCUCI + ' Port:' + RPCBroker1.ANURPCPort +

')';

labeldivision.caption := '';

If ESSOVersion = True Then

Begin

labeldivision.visible := true;

RPCBroker1.RemoteProcedure := 'DVBAB SET DIVISION';

RPCBroker1.Param[1].Value := RemoteUserDivisionNumber;

RPCBroker1.Param[1].PType := literal;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB SET DIVISION could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB SET DIVISION could not be accessed!');

End;

End;

Try

If piece(rpcbroker1.results[0], '^', 1) = '0' Then

Begin

LabelDivision.Caption := 'Division: Set division failed!';

ShowmessageCAPRI('WARNING: There was a problem setting the user''s division. Some reports may not function correctly. ' + piece(rpcbroker1.results[0], '^', 2));

End

Else

Begin

// User's division IEN is in piece 1, but we don't need it right now.

LabelDivision.caption := 'Division: ' + piece(rpcbroker1.results[0], '^', 2);

End;

Except

Begin

End;

End;

End

Else

Begin

// Non-remote mode

labelDivision.Caption := 'Division: ' + UserDivision;

End;

PanelDivision.Width := labelDivision.Width + 8;

If labelDivision.Caption = '' Then

Begin

ShowmessageCAPRI('WARNING: There was a problem setting the user''s division. Please check the host system and log in again. If the problem continues, notify your IT support service.');

timerhalt.enabled := true;

End;

Panel1.Visible := True;

SwitchToSite := ''; //Clear this, just in case it was previously set

AnimateLogo(False);

If RPCBroker1.Connected = False Then

Begin

frmMain.Caption := 'CAPRI not Connected'

End;

End

Else

Begin

TimeOutVal := 0;

TimeOutTimer.Enabled := False;

RPCBroker1.Connected := False;

ClearForm; //CodeCR248 - rpm 3/8/12 - clear the header and tab set

Panel1.Visible := False;

AnimateLogo(False);

End;

If Assigned(frmSplashScreen) Then

Begin

frmSplashScreen.Hide;

frmSplashScreen.release;

frmSplashScreen := Nil;

End;

If Assigned(frmAlerts) Then

Begin

If (frmAlerts.buttonUnsigned.enabled = true) Or (frmAlerts.buttonReview.enabled = true)

Or (frmAlerts.buttonUncosigned.enabled = true) Or (frmAlerts.buttonResolve.enabled = true)

Or (frmAlerts.btnManifests.enabled = true) // 2014-11-03 LMS Fix bug to show alert dialog when manifests are available

Then

Begin

frmAlerts.ShowModal;

End;

frmAlerts.Hide;

frmAlerts.release;

frmAlerts := Nil;

End;

If frmMain.Visible Then

//SelectPatient1Click(Sender);

actFileSelectPatientExecute(Sender);

For x := 1 To 100 Do

application.processmessages;

If Assigned(frmTIUCosign) Then

Begin

FreeAndNil(frmTIUCosign);

End;

If Assigned(frmUncosigned) Then

Begin

frmUncosigned.Hide;

Try

frmUncosigned.release;

Except

End;

frmUncosigned := Nil;

End;

If UnsignedView <> Nil Then

Begin

Try

UnsignedView.Release;

Except

End;

UnsignedView := Nil;

End;

switchtosite := '';

ContextorChangeMessage := '';

CCOWBreakLink := False;

// Check if CPEP has been notified that this used has used this version before

// Only do it one time.

tempstring2 := version;

If pos('\*', tempstring2) > 0 Then

Repeat

tempstring2 := copy(tempstring2, 1, pos('\*', tempstring2) - 1) + '\_' + copy(tempstring2, pos('\*', tempstring2) + 1, 255);

Until pos('\*', tempstring2) = 0;

If Not fileexists(tempdir + tempstring2 + '.validated') Then

Begin

tempstring := '';

tempstring := authorname + '^' + UserDivision + '^' + UserDivisionNumber + '^' + version;

aStream := TMemoryStream.Create;

formNews.GetURL('http://DNS cpepa/cgi-bin/capri\_track.pl?' + tempstring, aStream);

formNews.RichEditNews.Lines.LoadFromStream(aStream);

formNews.RichEditNews.Lines.SaveToFile(tempdir + tempstring2 + '.validated');

formNews.RichEditNews.Lines.Clear;

formNews.GetURL('http://' + newsURL + 'NewsHeaders.txt', aStream);

formNews.RichEditNews.Lines.LoadFromStream(aStream);

aStream := nil;

End;

End;

Table 48: Functions

| Function Name | btnAddRequestClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modify routine to handle new Request Status database field update. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnAddRequestClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

Procedure TfrmMain.btnAddRequestClick(Sender: TObject);

Var

x, y: integer;

MyTreeNode: Array[0..50000] Of TTreeNode;

NodeCount: LongInt;

temptextz: String;

Begin

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

frmAddress := TfrmAddress.Create(frmMain);

frmAddress.CountyFake.Text := '';

frmAddress.Top := 7 + frmMain.Top + ((frmMain.Height - frmAddress.Height) Div 2);

frmAddress.Left := frmMain.Left + ((frmMain.Width - frmAddress.Width) Div 2);

frmAddress.FMGetsPtAddress.IENS := PatientIEN;

frmAddress.FMGetsPtAddress.GetAndFill;

frmAddress.PopulateCountry;

If frmAddress.FMEditCounty.Text <> '' Then

Begin

For x := 0 To FMCountiesList.Items.Count - 1 Do

Begin

If Copy(FMCountiesList.Items[x], Pos(' ', FMCountiesList.Items[x]) + 4, 128) = frmAddress.FMEditCounty.Text Then

Begin

FMCountiesList.ItemIndex := x;

frmAddress.CountyFake.Text := FMCountiesList.Items[x];

End;

End;

End;

fmListerStates.GetList(frmAddress.FMStatesList.Items);

If FMEditState.Text <> '' Then

Begin

For x := 0 To frmAddress.FMStatesList.Items.Count - 1 Do // Find Selected State

If frmAddress.FMStatesList.Items[x] = FMEditState.Text Then

frmAddress.FMStatesList.ItemIndex := x;

//Now find selected county

FMListerCounties.IENS := ',' + frmAddress.FMStatesList.GetSelectedRecord.IEN;

fmListerCounties.GetList(frmAddress.FMCountiesList.Items);

If FMEditCounty.Text <> '' Then

Begin

For x := 0 To frmAddress.FMCountiesList.Items.Count - 1 Do

Begin

If Copy(frmAddress.FMCountiesList.Items[x], Pos(' ', frmAddress.FMCountiesList.Items[x]) + 4, 128) = FMEditCounty.Text Then

Begin

frmAddress.FMCountiesList.ItemIndex := x;

CountyFake.Text := frmAddress.FMCountiesList.Items[x];

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

End;

End;

End;

ContextorChangeMessage := 'You are editing an address. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

addressinfo := '';

If frmAddress.ShowModal = mrOK Then

Begin

temptextz := 'Address: ';

If (frmAddress.FMEditStreet1.Text <> '') And (frmAddress.FMEditStreet1.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet1.Text + CR + LF;

If (frmAddress.FMEditStreet1.Text <> '') And (frmAddress.FMEditStreet1.Text <> '@') Then

temptextz := ' ';

If (frmAddress.FMEditStreet2.Text <> '') And (frmAddress.FMEditStreet2.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet2.Text + CR + LF;

If (frmAddress.FMEditStreet2.Text <> '') And (frmAddress.FMEditStreet2.Text <> '@') Then

temptextz := ' ';

If (frmAddress.FMEditStreet3.Text <> '') And (frmAddress.FMEditStreet3.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet3.Text + CR + LF;

If (frmAddress.FMEditCity.Text <> '') And (frmAddress.FMEditCity.Text <> '@') Then

addressinfo := addressinfo + 'City: ' + frmAddress.FMEditCity.Text + CR + LF;

If (frmAddress.StateFake.Text <> '') and (frmAddress.StateFake.Text <> '@') then

addressinfo := addressinfo + 'State: ' + frmAddress.StateFake.Text + CR + LF;

If (frmAddress.FMEditZip.Text <> '') And (frmAddress.FMEditZip.Text <> '@') Then

addressinfo := addressinfo + 'Zip: ' + frmAddress.FMEditZip.Text + CR + LF;

If (frmAddress.CountyFake.Text <> '') and (frmAddress.CountyFake.Text <> '@') Then

addressinfo := addressinfo + 'County: ' + frmAddress.CountyFake.Text + CR + LF;

if (frmAddress.FMEditProvince.Text <> '') and (frmAddress.FMEditProvince.Text <> '@') then

addressinfo := addressinfo + 'Province: ' + frmAddress.FMEditProvince.Text + CR + LF;

if (frmAddress.FMEditPostalCode.Text <> '') and (frmAddress.FMEditPostalCode.Text <> '@') then

addressinfo := addressinfo + 'Postal Code: ' + frmAddress.FMEditPostalCode.Text + CR + LF;

if (frmAddress.cboCountry.Text <> '') then

addressinfo := addressinfo + 'Country: ' + frmAddress.cboCountry.Text + CR + LF;

If (frmAddress.FMEditPhone.Text <> '') And (frmAddress.FMEditPhone.Text <> '@') Then

addressinfo := addressinfo + 'Phone: ' + frmAddress.FMEditPhone.Text + CR + LF;

If (frmAddress.FMEditOffice.Text <> '') And (frmAddress.FMEditOffice.Text <> '@') Then

addressinfo := addressinfo + 'Office Phone: ' + frmAddress.FMEditOffice.Text + CR + LF;

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR694 JRL 3/27/15

begin // CodeCR694 JRL 3/27/15

If (frmAddress.FMEditCell.Text <> '') And (frmAddress.FMEditCell.Text <> '@') Then // CodeCR694 JRL 3/13/15

addressinfo := addressinfo + 'Cell Phone: ' + frmAddress.FMEditCell.Text + CR + LF; // CodeCR694 JRL 3/13/15

If (frmAddress.FMEditEmail.Text <> '') And (frmAddress.FMEditEmail.Text <> '@') Then // CodeCR694 JRL 3/13/15

addressinfo := addressinfo + 'Email: ' + frmAddress.FMEditEmail.Text + CR + LF; // CodeCR694 JRL 3/13/15

end; // CodeCR694 JRL 3/27/15

If (frmAddress.AddressDate.Text <> '') Then

// addressinfo := addressinfo + CR + LF + frmAddress.Label12.Caption + ' ' + frmAddress.AddressDate.Text + CR + LF;

addressinfo := addressinfo + CR + LF + frmAddress.lblAddressDate.Caption + ' ' + frmAddress.AddressDate.Text + CR + LF; // rpk 4/20/2015

End;

(\*

If frmAddress.Showmodal=mrCancel then begin

Page95Control1.ActivePage:=AdminDocuments;

Tab95Control3.Tabs[Tab95Control3.TabIndex]:='Address';

ScrollBoxAddress.VertScrollBar.Position:=0;

frmAddress.release; frmAddress := nil;

ContextorChangeMessage:='';

CCOWBreakLink:=False;

exit;

end;

\*)

frmAddress.release;

frmAddress := Nil;

AddRequestIEN := '+1';

frmMain.RPCBroker1.Results.Clear;

RPCBroker1.RemoteProcedure := 'DVBAB CHECK CREDENTIALS';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB CHECK CREDENTIALS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('RPC DVBAB CHECK CREDENTIALS could not be accessed!');

End

End;

If RPCBroker1.Results[0] <> '[OK]' Then

Begin

ShowMessageCAPRI(RPCBroker1.Results[0]);

application.terminate;

Exit;

End;

btnAddRequest.Enabled := False;

btnViewExam.Enabled := False;

btnExamFinalReport.Enabled := False;

btnGenerateReport.Enabled := False;

btnGenerate7131Report.enabled := False;

frmNewExam := TfrmNewExam.Create(frmMain);

// If credentials are ok, then continue

// Populate lists and controls

//FMListerExams.GetList(frmNewExam.FMExamsList.Items);

//frmNewExam := TfrmNewExam.Create(frmMain);

frmNewExam.FMExamsList.Items := ListExams;

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

GetActiveCapriDivisions(listMedicalCenterDivision, frmNewExam.FMRoutingLocation.Items); //CodeCR173 - rpm 4/18/11

frmNewExam.FMListBoxBodySystems.GetList;

frmNewExam.TreeViewExams.Items.Clear;

nodecount := 0;

If frmNewExam.FMListBoxBodySystems.Items.Count > 0 Then

For x := 0 To frmNewExam.FMListBoxBodySystems.Items.Count - 1 Do

Begin

MyTreeNode[x] := frmNewExam.TreeViewExams.Items.Add(Nil, frmNewExam.FMListBoxBodySystems.Items[x]);

nodecount := nodecount + 1;

frmNewExam.FMListBoxBodySystems.ItemIndex := x;

For y := 0 To frmNewExam.FMExamsList.Items.Count - 1 Do

If Pos(' I ', frmNewExam.FMExamsList.Items[y]) = 0 Then

If Pos(' ' + frmNewExam.FMListBoxBodySystems.GetSelectedRecord.IEN + ' ', frmNewExam.FMExamsList.Items[y] + ' ') > 0 Then

Begin

MyTreeNode[NodeCount] := frmNewExam.TreeViewExams.Items.AddChild(MyTreeNode[x], Copy(frmNewExam.FMExamsList.Items[y], 1, Pos(' A', frmNewExam.FMExamsList.Items[y]) - 3));

inc(nodecount);

End;

If MyTreeNode[x].getFirstChild = Nil Then

MyTreeNode[x].Delete;

End;

//FMListerInstitutionFile.GetList(frmNewExam.FMRegionalOfficeNumber.Items);

frmNewExam.FMRegionalOfficeNumber.Items.Clear;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmNewExam.FMRegionalOfficeNumber.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

(\*

frmNewExam.FMRegionalOfficeNumber.Items:=lstInstitutions.Items;

If frmNewExam.FMRegionalOfficeNumber.Items.Count>0 then

For x:=frmNewExam.FMRegionalOfficeNumber.Items.Count-1 downto 0 do

If Pos('-RO ',frmNewExam.FMRegionalOfficeNumber.Items[x])=0 then

frmNewExam.FMRegionalOfficeNumber.Items.Delete(x) else

frmNewExam.FMRegionalOfficeNumber.Items[x]:=Copy(frmNewExam.FMRegionalOfficeNumber.Items[x],1,Pos(' ',frmNewExam.FMRegionalOfficeNumber.Items[x])-1);

\*)

frmNewExam.FMPatientName.Text := Copy(Panel1.Caption, 1, Pos(' SSN#', Panel1.Caption));

frmNewExam.FMPatientNameIEN.Text := '`' + PatientIEN;

frmNewExam.FMRequestedBy.Text := FMUserName.Caption;

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

frmNewExam.FMRequestDate.Text := 'NOW';

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

frmNewExam.FMRequestStatus.Text := 'N';

frmNewExam.FMClaimFolderRequired.Text := '';

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 3/26/15

frmNewExam.FMElectronicClaimFolder.Text := ''; // CodeCR704 JRL 3/16/15

frmNewExam.FMOtherDocumentsRequired.Text := '';

frmNewExam.FMRegionalOfficeNumber.Text := '';

frmNewExam.FMRegionalOfficeNumber.Enabled := True;

frmNewExam.FMPriorityOfExam.Text := '';

frmNewExam.FMPriorityOfExam.Enabled := True; // In case insufficient exam action turned it off

frmNewExam.FMOtherDisabilities1.Text := '';

frmNewExam.FMOtherDisabilities2.Text := '';

frmNewExam.FMOtherDisabilities3.Text := '';

frmNewExam.FMRoutingLocation.Text := '';

frmNewExam.FMLastRatingExamDate.Text := '';

frmNewExam.FMCommentsMemo.Lines.Clear;

//Setup for new entry

frmNewExam.FMPatientNameIEN.IENS := '+1';

frmNewExam.FMRequestDate.IENS := '+1';

frmNewExam.FMClaimFolderRequiredFake.IENS := '+1';

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 3/26/15

frmNewExam.FMElectronicClaimFolderFake.IENS := '+1'; // CodeCR704 JRL 3/16/15

frmNewExam.FMOtherDocumentsRequiredFake.IENS := '+1';

frmNewExam.FMRegionalOfficeNumberIEN.IENS := '+1';

frmNewExam.FMRequestedByIEN.IENS := '+1';

frmNewExam.FMRequestStatus.IENS := '+1';

frmNewExam.FMPriorityOfExamFake.IENS := '+1';

frmNewExam.FMOtherDisabilities1.IENS := '+1';

frmNewExam.FMOtherDisabilities2.IENS := '+1';

frmNewExam.FMOtherDisabilities3.IENS := '+1';

frmNewExam.FMRoutingLocationFake.IENS := '+1';

frmNewExam.FMLastRatingExamDate.IENS := '+1';

frmNewExam.FMCommentsMemo.IENS := '+1';

frmNewExam.FMOriginal2507Request.IENS := '+1';

frmNewExam.FMOriginal2507ProcessingTime.IENS := '+1';

frmNewExam.FMOriginal2507Request.Text := '';

frmNewExam.FMOriginal2507ProcessingTime.Text := '';

//Enter request

frmNewExam.Height := frmMain.Height - 32;

If frmNewExam.Height > 698 Then

frmNewExam.Height := 698;

frmNewExam.Top := 7 + frmMain.Top + ((frmMain.Height - frmNewExam.Height) Div 2);

frmNewExam.Left := frmMain.Left + ((frmMain.Width - frmNewExam.Width) Div 2);

frmNewExam.btnSendRequest.Top := frmNewExam.Height - 81; //73

frmNewExam.btnCancel.Top := frmNewExam.Height - 81; //73

frmNewExam.Scrollbox1.Height := frmNewExam.Height - 82; //-74

//frmMain.BorderIcons:=frmMain.BorderIcons-[biSystemMenu];

frmNewExam.Scrollbox1.VertScrollBar.Position := 0;

frmNewExam.FMListerCustomExamListDivisionsListBox.GetList;

//Check for inactive divisions

If frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For x := 0 To frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmNewExam.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

frmNewExam.CustomExamListDivisions.IENS := frmNewExam.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmNewExam.CustomExamListDivisions.GetAndFill;

End;

frmNewExam.FMRoutingLocation.ItemIndex := -1;

//rra 970009 - set Routing Location Information to null until a Routing Location is selected

frmNewExam.FMMemoDivisionComments.text := '';

// Set up division...

searchforstring := UserDivision;

frmNewExam.FMRegionalOfficeNumberEnter(Application);

searchforstring := '';

If frmNewExam.FMRegionalOfficeNumberIEN.Text <> '' Then

frmNewExam.FMRegionalOfficeNumber.Enabled := False;

(\*

frmNewExam.FMRegionalOfficeNumber.Enabled:=False;

frmNewExam.FMRegionalOfficeNumber.Text:=UserDivision;

frmNewExam.FMRegionalOfficeNumberExit(Application);

\*)

frmNewExam.Panel1.Visible := False;

ContextorChangeMessage := 'You are adding an exam request. If you continue, you will lose this work.';

CCOWBreakLink := True;

frmNewExam.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmNewExam.release;

frmNewExam := Nil;

Try

frmROFinder.Hide;

Except

End;

//frmMain.BorderIcons:=frmMain.BorderIcons+[biSystemMenu];

ExamRequestRefreshClick(Application);

If ReadOnlyMode = False Then

btnAddRequest.Enabled := True;

If FMExamRequestListbox.ItemIndex > -1 Then

Begin

btnViewExam.Enabled := True;

btnExamFinalReport.Enabled := True;

btnGenerateReport.Enabled := True;

End;

End;

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

Procedure TfrmMain.btnAddRequestClick(Sender: TObject);

Var

x, y: integer;

MyTreeNode: Array[0..50000] Of TTreeNode;

NodeCount: LongInt;

temptextz: String;

Begin

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

frmAddress := TfrmAddress.Create(frmMain);

frmAddress.CountyFake.Text := '';

frmAddress.Top := 7 + frmMain.Top + ((frmMain.Height - frmAddress.Height) Div 2);

frmAddress.Left := frmMain.Left + ((frmMain.Width - frmAddress.Width) Div 2);

frmAddress.FMGetsPtAddress.IENS := PatientIEN;

frmAddress.FMGetsPtAddress.GetAndFill;

frmAddress.PopulateCountry;

If frmAddress.FMEditCounty.Text <> '' Then

Begin

For x := 0 To FMCountiesList.Items.Count - 1 Do

Begin

If Copy(FMCountiesList.Items[x], Pos(' ', FMCountiesList.Items[x]) + 4, 128) = frmAddress.FMEditCounty.Text Then

Begin

FMCountiesList.ItemIndex := x;

frmAddress.CountyFake.Text := FMCountiesList.Items[x];

End;

End;

End;

fmListerStates.GetList(frmAddress.FMStatesList.Items);

If FMEditState.Text <> '' Then

Begin

For x := 0 To frmAddress.FMStatesList.Items.Count - 1 Do // Find Selected State

If frmAddress.FMStatesList.Items[x] = FMEditState.Text Then

frmAddress.FMStatesList.ItemIndex := x;

//Now find selected county

FMListerCounties.IENS := ',' + frmAddress.FMStatesList.GetSelectedRecord.IEN;

fmListerCounties.GetList(frmAddress.FMCountiesList.Items);

If FMEditCounty.Text <> '' Then

Begin

For x := 0 To frmAddress.FMCountiesList.Items.Count - 1 Do

Begin

If Copy(frmAddress.FMCountiesList.Items[x], Pos(' ', frmAddress.FMCountiesList.Items[x]) + 4, 128) = FMEditCounty.Text Then

Begin

frmAddress.FMCountiesList.ItemIndex := x;

CountyFake.Text := frmAddress.FMCountiesList.Items[x];

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

End;

End;

End;

ContextorChangeMessage := 'You are editing an address. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

addressinfo := '';

If frmAddress.ShowModal = mrOK Then

Begin

temptextz := 'Address: ';

If (frmAddress.FMEditStreet1.Text <> '') And (frmAddress.FMEditStreet1.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet1.Text + CR + LF;

If (frmAddress.FMEditStreet1.Text <> '') And (frmAddress.FMEditStreet1.Text <> '@') Then

temptextz := ' ';

If (frmAddress.FMEditStreet2.Text <> '') And (frmAddress.FMEditStreet2.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet2.Text + CR + LF;

If (frmAddress.FMEditStreet2.Text <> '') And (frmAddress.FMEditStreet2.Text <> '@') Then

temptextz := ' ';

If (frmAddress.FMEditStreet3.Text <> '') And (frmAddress.FMEditStreet3.Text <> '@') Then

addressinfo := addressinfo + temptextz + frmAddress.FMEditStreet3.Text + CR + LF;

If (frmAddress.FMEditCity.Text <> '') And (frmAddress.FMEditCity.Text <> '@') Then

addressinfo := addressinfo + 'City: ' + frmAddress.FMEditCity.Text + CR + LF;

If (frmAddress.StateFake.Text <> '') and (frmAddress.StateFake.Text <> '@') then

addressinfo := addressinfo + 'State: ' + frmAddress.StateFake.Text + CR + LF;

If (frmAddress.FMEditZip.Text <> '') And (frmAddress.FMEditZip.Text <> '@') Then

addressinfo := addressinfo + 'Zip: ' + frmAddress.FMEditZip.Text + CR + LF;

If (frmAddress.CountyFake.Text <> '') and (frmAddress.CountyFake.Text <> '@') Then

addressinfo := addressinfo + 'County: ' + frmAddress.CountyFake.Text + CR + LF;

if (frmAddress.FMEditProvince.Text <> '') and (frmAddress.FMEditProvince.Text <> '@') then

addressinfo := addressinfo + 'Province: ' + frmAddress.FMEditProvince.Text + CR + LF;

if (frmAddress.FMEditPostalCode.Text <> '') and (frmAddress.FMEditPostalCode.Text <> '@') then

addressinfo := addressinfo + 'Postal Code: ' + frmAddress.FMEditPostalCode.Text + CR + LF;

if (frmAddress.cboCountry.Text <> '') then

addressinfo := addressinfo + 'Country: ' + frmAddress.cboCountry.Text + CR + LF;

If (frmAddress.FMEditPhone.Text <> '') And (frmAddress.FMEditPhone.Text <> '@') Then

addressinfo := addressinfo + 'Phone: ' + frmAddress.FMEditPhone.Text + CR + LF;

If (frmAddress.FMEditOffice.Text <> '') And (frmAddress.FMEditOffice.Text <> '@') Then

addressinfo := addressinfo + 'Office Phone: ' + frmAddress.FMEditOffice.Text + CR + LF;

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR694 JRL 3/27/15

begin // CodeCR694 JRL 3/27/15

If (frmAddress.FMEditCell.Text <> '') And (frmAddress.FMEditCell.Text <> '@') Then // CodeCR694 JRL 3/13/15

addressinfo := addressinfo + 'Cell Phone: ' + frmAddress.FMEditCell.Text + CR + LF; // CodeCR694 JRL 3/13/15

If (frmAddress.FMEditEmail.Text <> '') And (frmAddress.FMEditEmail.Text <> '@') Then // CodeCR694 JRL 3/13/15

addressinfo := addressinfo + 'Email: ' + frmAddress.FMEditEmail.Text + CR + LF; // CodeCR694 JRL 3/13/15

end; // CodeCR694 JRL 3/27/15

If (frmAddress.AddressDate.Text <> '') Then

// addressinfo := addressinfo + CR + LF + frmAddress.Label12.Caption + ' ' + frmAddress.AddressDate.Text + CR + LF;

addressinfo := addressinfo + CR + LF + frmAddress.lblAddressDate.Caption + ' ' + frmAddress.AddressDate.Text + CR + LF; // rpk 4/20/2015

End;

(\*

If frmAddress.Showmodal=mrCancel then begin

Page95Control1.ActivePage:=AdminDocuments;

Tab95Control3.Tabs[Tab95Control3.TabIndex]:='Address';

ScrollBoxAddress.VertScrollBar.Position:=0;

frmAddress.release; frmAddress := nil;

ContextorChangeMessage:='';

CCOWBreakLink:=False;

exit;

end;

\*)

frmAddress.release;

frmAddress := Nil;

AddRequestIEN := '+1';

frmMain.RPCBroker1.Results.Clear;

RPCBroker1.RemoteProcedure := 'DVBAB CHECK CREDENTIALS';

RPCBrokerCall;

Try

RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB CHECK CREDENTIALS could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('RPC DVBAB CHECK CREDENTIALS could not be accessed!');

End

End;

If RPCBroker1.Results[0] <> '[OK]' Then

Begin

ShowMessageCAPRI(RPCBroker1.Results[0]);

application.terminate;

Exit;

End;

btnAddRequest.Enabled := False;

btnViewExam.Enabled := False;

btnExamFinalReport.Enabled := False;

btnGenerateReport.Enabled := False;

btnGenerate7131Report.enabled := False;

frmNewExam := TfrmNewExam.Create(frmMain);

// If credentials are ok, then continue

// Populate lists and controls

//FMListerExams.GetList(frmNewExam.FMExamsList.Items);

//frmNewExam := TfrmNewExam.Create(frmMain);

frmNewExam.FMExamsList.Items := ListExams;

frmNewExam.CheckBoxExamsToComplete.Items.Clear;

GetActiveCapriDivisions(listMedicalCenterDivision, frmNewExam.FMRoutingLocation.Items); //CodeCR173 - rpm 4/18/11

frmNewExam.FMListBoxBodySystems.GetList;

frmNewExam.TreeViewExams.Items.Clear;

nodecount := 0;

If frmNewExam.FMListBoxBodySystems.Items.Count > 0 Then

For x := 0 To frmNewExam.FMListBoxBodySystems.Items.Count - 1 Do

Begin

MyTreeNode[x] := frmNewExam.TreeViewExams.Items.Add(Nil, frmNewExam.FMListBoxBodySystems.Items[x]);

nodecount := nodecount + 1;

frmNewExam.FMListBoxBodySystems.ItemIndex := x;

For y := 0 To frmNewExam.FMExamsList.Items.Count - 1 Do

If Pos(' I ', frmNewExam.FMExamsList.Items[y]) = 0 Then

If Pos(' ' + frmNewExam.FMListBoxBodySystems.GetSelectedRecord.IEN + ' ', frmNewExam.FMExamsList.Items[y] + ' ') > 0 Then

Begin

MyTreeNode[NodeCount] := frmNewExam.TreeViewExams.Items.AddChild(MyTreeNode[x], Copy(frmNewExam.FMExamsList.Items[y], 1, Pos(' A', frmNewExam.FMExamsList.Items[y]) - 3));

inc(nodecount);

End;

If MyTreeNode[x].getFirstChild = Nil Then

MyTreeNode[x].Delete;

End;

//FMListerInstitutionFile.GetList(frmNewExam.FMRegionalOfficeNumber.Items);

frmNewExam.FMRegionalOfficeNumber.Items.Clear;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmNewExam.FMRegionalOfficeNumber.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

(\*

frmNewExam.FMRegionalOfficeNumber.Items:=lstInstitutions.Items;

If frmNewExam.FMRegionalOfficeNumber.Items.Count>0 then

For x:=frmNewExam.FMRegionalOfficeNumber.Items.Count-1 downto 0 do

If Pos('-RO ',frmNewExam.FMRegionalOfficeNumber.Items[x])=0 then

frmNewExam.FMRegionalOfficeNumber.Items.Delete(x) else

frmNewExam.FMRegionalOfficeNumber.Items[x]:=Copy(frmNewExam.FMRegionalOfficeNumber.Items[x],1,Pos(' ',frmNewExam.FMRegionalOfficeNumber.Items[x])-1);

\*)

frmNewExam.FMPatientName.Text := Copy(Panel1.Caption, 1, Pos(' SSN#', Panel1.Caption));

frmNewExam.FMPatientNameIEN.Text := '`' + PatientIEN;

frmNewExam.FMRequestedBy.Text := FMUserName.Caption;

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

frmNewExam.FMRequestDate.Text := 'NOW';

frmNewExam.FMRequestedByIEN.Text := '`' + AuthorIEN;

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

**frmNewExam.FMRequestStatus.Text := 'NEW' // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

**else // Patch 193 JRL 4/21/16 NEW REQUEST STATUS FIELD UPDATE**

frmNewExam.FMRequestStatus.Text := 'N';

frmNewExam.FMClaimFolderRequired.Text := '';

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 3/26/15

frmNewExam.FMElectronicClaimFolder.Text := ''; // CodeCR704 JRL 3/16/15

frmNewExam.FMOtherDocumentsRequired.Text := '';

frmNewExam.FMRegionalOfficeNumber.Text := '';

frmNewExam.FMRegionalOfficeNumber.Enabled := True;

frmNewExam.FMPriorityOfExam.Text := '';

frmNewExam.FMPriorityOfExam.Enabled := True; // In case insufficient exam action turned it off

frmNewExam.FMOtherDisabilities1.Text := '';

frmNewExam.FMOtherDisabilities2.Text := '';

frmNewExam.FMOtherDisabilities3.Text := '';

frmNewExam.FMRoutingLocation.Text := '';

frmNewExam.FMLastRatingExamDate.Text := '';

frmNewExam.FMCommentsMemo.Lines.Clear;

//Setup for new entry

frmNewExam.FMPatientNameIEN.IENS := '+1';

frmNewExam.FMRequestDate.IENS := '+1';

frmNewExam.FMClaimFolderRequiredFake.IENS := '+1';

If IsPatchInstalled('DVBA\*2.7\*190') then // CodeCR704 JRL 3/26/15

frmNewExam.FMElectronicClaimFolderFake.IENS := '+1'; // CodeCR704 JRL 3/16/15

frmNewExam.FMOtherDocumentsRequiredFake.IENS := '+1';

frmNewExam.FMRegionalOfficeNumberIEN.IENS := '+1';

frmNewExam.FMRequestedByIEN.IENS := '+1';

frmNewExam.FMRequestStatus.IENS := '+1';

frmNewExam.FMPriorityOfExamFake.IENS := '+1';

frmNewExam.FMOtherDisabilities1.IENS := '+1';

frmNewExam.FMOtherDisabilities2.IENS := '+1';

frmNewExam.FMOtherDisabilities3.IENS := '+1';

frmNewExam.FMRoutingLocationFake.IENS := '+1';

frmNewExam.FMLastRatingExamDate.IENS := '+1';

frmNewExam.FMCommentsMemo.IENS := '+1';

frmNewExam.FMOriginal2507Request.IENS := '+1';

frmNewExam.FMOriginal2507ProcessingTime.IENS := '+1';

frmNewExam.FMOriginal2507Request.Text := '';

frmNewExam.FMOriginal2507ProcessingTime.Text := '';

//Enter request

frmNewExam.Height := frmMain.Height - 32;

If frmNewExam.Height > 698 Then

frmNewExam.Height := 698;

frmNewExam.Top := 7 + frmMain.Top + ((frmMain.Height - frmNewExam.Height) Div 2);

frmNewExam.Left := frmMain.Left + ((frmMain.Width - frmNewExam.Width) Div 2);

frmNewExam.btnSendRequest.Top := frmNewExam.Height - 81; //73

frmNewExam.btnCancel.Top := frmNewExam.Height - 81; //73

frmNewExam.Scrollbox1.Height := frmNewExam.Height - 82; //-74

//frmMain.BorderIcons:=frmMain.BorderIcons-[biSystemMenu];

frmNewExam.Scrollbox1.VertScrollBar.Position := 0;

frmNewExam.FMListerCustomExamListDivisionsListBox.GetList;

//Check for inactive divisions

If frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For x := 0 To frmNewExam.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmNewExam.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

frmNewExam.CustomExamListDivisions.IENS := frmNewExam.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmNewExam.CustomExamListDivisions.GetAndFill;

End;

frmNewExam.FMRoutingLocation.ItemIndex := -1;

//rra 970009 - set Routing Location Information to null until a Routing Location is selected

frmNewExam.FMMemoDivisionComments.text := '';

// Set up division...

searchforstring := UserDivision;

frmNewExam.FMRegionalOfficeNumberEnter(Application);

searchforstring := '';

If frmNewExam.FMRegionalOfficeNumberIEN.Text <> '' Then

frmNewExam.FMRegionalOfficeNumber.Enabled := False;

(\*

frmNewExam.FMRegionalOfficeNumber.Enabled:=False;

frmNewExam.FMRegionalOfficeNumber.Text:=UserDivision;

frmNewExam.FMRegionalOfficeNumberExit(Application);

\*)

frmNewExam.Panel1.Visible := False;

ContextorChangeMessage := 'You are adding an exam request. If you continue, you will lose this work.';

CCOWBreakLink := True;

frmNewExam.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmNewExam.release;

frmNewExam := Nil;

Try

frmROFinder.Hide;

Except

End;

//frmMain.BorderIcons:=frmMain.BorderIcons+[biSystemMenu];

ExamRequestRefreshClick(Application);

If ReadOnlyMode = False Then

btnAddRequest.Enabled := True;

If FMExamRequestListbox.ItemIndex > -1 Then

Begin

btnViewExam.Enabled := True;

btnExamFinalReport.Enabled := True;

btnGenerateReport.Enabled := True;

End;

End;

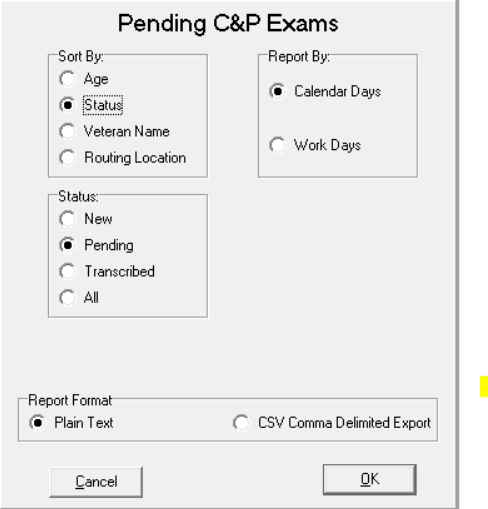
##### GUI – Reports.pas

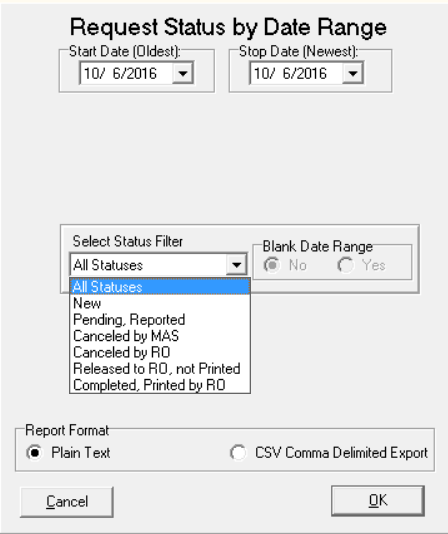
Table 36: GUI

| Unit Name | Description |
| --- | --- |
| Reports.pas | Modified two reports to integrate new Re-Route statuses. |

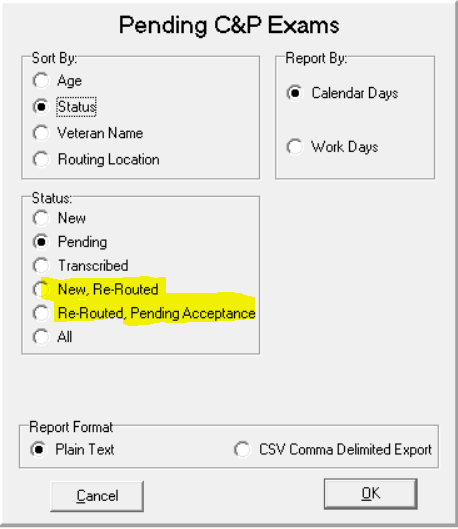
##### GUI Classes – N/A in Requirement 2.6.1

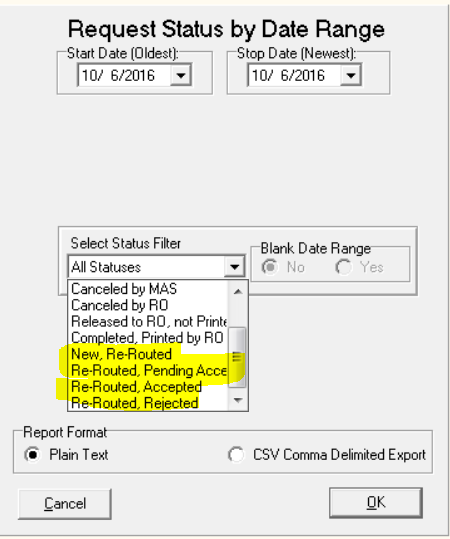
##### Current Form





##### Modified Form





##### Components on Form – N/A for Requirement 2.6.1

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| Button1Click | Event | Modified GUI and code to allow new Re-Routing statuses to be selected. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| GetRequestStatusReport | Procedure | Modified GUI and code to allow new Re-Routing statuses to be selected. |

##### Special References – N/A in Requirement 2.6.1

##### Class Events – N/A in Requirement 2.6.1

##### Class Methods – N/A in Requirement 2.6.1

##### Class Properties – N/A in Requirement 2.6.1

##### Uses Clause – N/A in Requirement 2.6.1

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | Reports | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** |  | | | |

| Current Form Layout |
| --- |
| See current form above |

| Modified Form Layout (Changes are in bold) |
| --- |
| See current form above |

##### Functions

Table 48: Functions

| Function Name | Button1Click | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified GUI and code to allow new Re-Routing statuses to be selected. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | Button1Click | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

//------------------------------------------------------------------------------

// Pending C&P Exams Report

//------------------------------------------------------------------------------

// RPC: DVBAB PENDING C&P REPORT

// Input Parameters:

// 0 : Sort By

// 'A’ – Age, 'S’ – Status, 'V’ – Veteran Name, 'R’ – Routing Location

// 1 : If Sort By = 'S' (Status)

// 'N’ – New, 'P’ – Pending, Reported, 'T’ – Transcribed, 'NR’ – New, Re-Routed

// 'RP’ – Re-Routed, Pending Acceptance, 'RA’ – Re-Routed, Accepted,

// 'RR’ – Re-Routed, Rejected, 'A’ – All Statuses

// 2 : if Sort By = 'A' (Age)

// Earliest Days (edit1.text)

// 3 : if Sort By = 'A' (Age)

// Oldest Days (edit2.text)

// 4 : if Sort By = 'R' (Routing Location)

// FMRoutingLocation selected

// 5 : Report By (always set, not dependent upon Sorting)

// 'C' - Calendar Days, 'W' - Work Days

// 6 : Report Format

// '0' - Plain Text, '1' - Delimited

// 7 : Report Mode \*\* Always '0' from this location

// '0' - Local Mode

// '1' - Remote Consolidate Reports mode (frmRemoteReports.PendingCnP)

//------------------------------------------------------------------------------

Procedure Tfrmreports.Button1Click(Sender: TObject);

var

ADelimiter: String; // CodeCR470 JRL 4/2/13

Filename: String; // CodeCR470 JRL 4/3/13

DelimitedLines: TStringList; // CodeCR470 JRL 4/3/13

AbbrevReportName: String; // CodeCR470 JRL 4/3/13

StatusSelection : String; // Patch 193 JRL 7/25/16

Begin

If RadioGroup1.ItemIndex = -1 Then

Begin

ShowMessageCAPRI('Please select a type of sort for this report.');

exit;

End;

If Radiogroup1.Items[RadioGroup1.ItemIndex] = 'Age' Then

Begin

If StrToInt(Edit1.Text) > StrToInt(Edit2.Text) Then

Begin

ShowMessageCAPRI('Earliest age must be less than oldest age.');

exit;

End;

End;

If Radiogroup1.Items[RadioGroup1.ItemIndex] = 'Routing Location' Then

Begin

If FMRoutingLocationFake.Text = '' Then

Begin

ShowMessageCAPRI('Please select a routing location.');

exit;

End;

End;

If label1\_3.caption = 'Pending C&&P Exams' Then

Begin

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := 'DVBAB PENDING C&P REPORT';

// Always set sort by

frmMain.RPCBroker1.Param[0].Value := RadioGroup1.Items[RadioGroup1.ItemIndex];

frmMain.RPCBroker1.Param[0].Value := frmMain.RPCBroker1.Param[0].Value[1]; // Just get 1st char

frmMain.RPCBroker1.Param[0].PType := literal;

// Status

frmMain.RPCBroker1.Param[1].Value := '';

frmMain.RPCBroker1.Param[1].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'S' Then // Status

Begin

frmMain.RPCBroker1.Param[1].Value := frmMain.RPCBroker1.Param[1].Value[1]; // Just get 1st char

End;

frmMain.RPCBroker1.Param[2].Value := '';

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[3].Value := '';

frmMain.RPCBroker1.Param[3].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'A' Then

Begin

frmMain.RPCBroker1.Param[2].Value := Edit1.Text;

frmMain.RPCBroker1.Param[3].Value := Edit2.Text;

End;

frmMain.RPCBroker1.Param[4].Value := '';

frmMain.RPCBroker1.Param[4].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'R' Then

Begin

frmMain.RPCBroker1.Param[4].Value := FMRoutingLocationFake.Text;

End;

// Always set (c)alendar or (w)ork days

frmMain.RPCBroker1.Param[5].Value := '';

frmMain.RPCBroker1.Param[5].PType := literal;

frmMain.RPCBroker1.Param[5].Value := RadioGroup3.Items[RadioGroup3.ItemIndex];

frmMain.RPCBroker1.Param[5].Value := frmMain.RPCBroker1.Param[5].Value[1]; // Just get 1st char

// Delimited or not

case rgPendingCnPRptFmt.ItemIndex of // CodeCR470 JRL 4/5/13

0: ADelimiter := '0'; // CodeCR470 JRL 4/5/13

1: ADelimiter := '1'; // CodeCR470 JRL 4/5/13

end; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[6].Value := ADelimiter; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[6].PType := literal; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[7].Value := '0'; // Local mode vs. // CodeCR470 JRL 5/22/13

frmMain.RPCBroker1.Param[7].PType := literal; // Consolidated remote rpts // CodeCR470 JRL 5/22/13

try

DisplayBusyAnimation(True);

frmMain.RPCBroker1.Call;

except on E: Exception do

begin

ShowMessageCAPRI('DVBAB PENDING C&P REPORT returned an error!' + #13#10 + E.Message);

Screen.Cursor := crDefault;

exit;

end;

end;

panelReportChoice3.Visible := False;

ReportMemo.Visible := False;

If frmMain.RPCBroker1.Results.Count > 0 Then

Begin

if ADelimiter = '0' then // normal report // CodeCR470 JRL 4/5/13

begin

Repeat

Begin

Screen.Cursor := crHourglass;

If Pos('^', frmMain.RPCBroker1.Results[0]) > 0 Then

Begin

ReportMemo.Lines.Add(Copy(frmMain.RPCBroker1.Results[0], 1, Pos('^', frmMain.RPCBroker1.Results[0]) - 1));

frmMain.RPCBroker1.Results[0] := Copy(frmMain.RPCBroker1.Results[0], Pos('^', frmMain.RPCBroker1.Results[0]) + 1, Length(frmMain.RPCBroker1.Results[0]));

End

Else

Begin

ReportMemo.Lines.Add(frmMain.RPCBroker1.Results[0]);

frmMain.RPCBroker1.Results[0] := '';

End;

End

Until frmMain.RPCBroker1.Results[0] = '';

end

else // delimited report CodeCR470 JRL 4/5/13

begin

AbbrevReportName := 'PendingCPExam'; // CodeCR470 JRL 4/3/13

DelimitedLines := TStringList.Create; // CodeCR470 JRL 4/3/13

try

CAPRIQuickCopy(frmMain.RPCBroker1.Results, DelimitedLines);

FileName := SaveDelimitedReportCSV(DelimitedLines, AbbrevReportName

+ '\_'

+ FormatDateTime('YYYYMMDD', dtpInsuffStart.Date)

+ '\_'

+ FormatDateTime('YYYYMMDD', dtpInsuffEnd.Date));

ReportMemo.Lines.Insert(0, '');

if (FileName <> '') then

ReportMemo.Lines.Insert(0, IntToStr(DelimitedLines.Count)

+ ' Extract lines saved to: ' + #13#10

+ FileName)

else

ReportMemo.Lines.Insert(0, 'No delimited text lines were saved!');

finally

FreeAndNil(DelimitedLines);

end;

end;

Screen.Cursor := crDefault;

End;

If ReportMemo.Lines.count = 0 Then

ReportMemo.Lines.Add('No data found for requested report criteria.');

ReportMemo.Visible := True;

ReportMemo.SelStart := 0;

ReportMemo.SelLength := 0;

ReportMemo.SetFocus;

End;

panelReportChoice3.Visible := False;

End;

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

//------------------------------------------------------------------------------

// Pending C&P Exams Report

//------------------------------------------------------------------------------

// RPC: DVBAB PENDING C&P REPORT

// Input Parameters:

// 0 : Sort By

// 'A’ – Age, 'S’ – Status, 'V’ – Veteran Name, 'R’ – Routing Location

// 1 : If Sort By = 'S' (Status)

// 'N’ – New, 'P’ – Pending, Reported, 'T’ – Transcribed, 'NR’ – New, Re-Routed

// 'RP’ – Re-Routed, Pending Acceptance, 'RA’ – Re-Routed, Accepted,

// 'RR’ – Re-Routed, Rejected, 'A’ – All Statuses

// 2 : if Sort By = 'A' (Age)

// Earliest Days (edit1.text)

// 3 : if Sort By = 'A' (Age)

// Oldest Days (edit2.text)

// 4 : if Sort By = 'R' (Routing Location)

// FMRoutingLocation selected

// 5 : Report By (always set, not dependent upon Sorting)

// 'C' - Calendar Days, 'W' - Work Days

// 6 : Report Format

// '0' - Plain Text, '1' - Delimited

// 7 : Report Mode \*\* Always '0' from this location

// '0' - Local Mode

// '1' - Remote Consolidate Reports mode (frmRemoteReports.PendingCnP)

//------------------------------------------------------------------------------

Procedure Tfrmreports.Button1Click(Sender: TObject);

var

ADelimiter: String; // CodeCR470 JRL 4/2/13

Filename: String; // CodeCR470 JRL 4/3/13

DelimitedLines: TStringList; // CodeCR470 JRL 4/3/13

AbbrevReportName: String; // CodeCR470 JRL 4/3/13

StatusSelection : String; // Patch 193 JRL 7/25/16

Begin

If RadioGroup1.ItemIndex = -1 Then

Begin

ShowMessageCAPRI('Please select a type of sort for this report.');

exit;

End;

If Radiogroup1.Items[RadioGroup1.ItemIndex] = 'Age' Then

Begin

If StrToInt(Edit1.Text) > StrToInt(Edit2.Text) Then

Begin

ShowMessageCAPRI('Earliest age must be less than oldest age.');

exit;

End;

End;

If Radiogroup1.Items[RadioGroup1.ItemIndex] = 'Routing Location' Then

Begin

If FMRoutingLocationFake.Text = '' Then

Begin

ShowMessageCAPRI('Please select a routing location.');

exit;

End;

End;

If label1\_3.caption = 'Pending C&&P Exams' Then

Begin

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := 'DVBAB PENDING C&P REPORT';

// Always set sort by

frmMain.RPCBroker1.Param[0].Value := RadioGroup1.Items[RadioGroup1.ItemIndex];

frmMain.RPCBroker1.Param[0].Value := frmMain.RPCBroker1.Param[0].Value[1]; // Just get 1st char

frmMain.RPCBroker1.Param[0].PType := literal;

// Status

frmMain.RPCBroker1.Param[1].Value := '';

frmMain.RPCBroker1.Param[1].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'S' Then // Status

Begin

**// Patch 193 JRL 7/25/16. Added in new statuses. Changed code (which**

**// originally chose the first character of the status) to choose the**

**// status code below.**

**case RadioGroup2.ItemIndex of**

**0: StatusSelection := 'N'; // New**

**1: StatusSelection := 'P'; // Pending**

**2: StatusSelection := 'T'; // Transcribed**

**3: StatusSelection := 'NR'; // New, Re-Routed**

**4: StatusSelection := 'RP'; // Re-Routed, Pending Acceptance**

**5: StatusSelection := 'A'; // All**

**end; // case**

**frmMain.RPCBroker1.Param[1].Value := RadioGroup2.Items[RadioGroup2.ItemIndex];**

**frmMain.RPCBroker1.Param[1].Value := StatusSelection; // Patch 193 JRL 7/25/16**

**// frmMain.RPCBroker1.Param[1].Value := frmMain.RPCBroker1.Param[1].Value[1]; // Just get 1st char**

End;

frmMain.RPCBroker1.Param[2].Value := '';

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[3].Value := '';

frmMain.RPCBroker1.Param[3].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'A' Then

Begin

frmMain.RPCBroker1.Param[2].Value := Edit1.Text;

frmMain.RPCBroker1.Param[3].Value := Edit2.Text;

End;

frmMain.RPCBroker1.Param[4].Value := '';

frmMain.RPCBroker1.Param[4].PType := literal;

If frmMain.RPCBroker1.Param[0].Value = 'R' Then

Begin

frmMain.RPCBroker1.Param[4].Value := FMRoutingLocationFake.Text;

End;

// Always set (c)alendar or (w)ork days

frmMain.RPCBroker1.Param[5].Value := '';

frmMain.RPCBroker1.Param[5].PType := literal;

frmMain.RPCBroker1.Param[5].Value := RadioGroup3.Items[RadioGroup3.ItemIndex];

frmMain.RPCBroker1.Param[5].Value := frmMain.RPCBroker1.Param[5].Value[1]; // Just get 1st char

// Delimited or not

case rgPendingCnPRptFmt.ItemIndex of // CodeCR470 JRL 4/5/13

0: ADelimiter := '0'; // CodeCR470 JRL 4/5/13

1: ADelimiter := '1'; // CodeCR470 JRL 4/5/13

end; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[6].Value := ADelimiter; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[6].PType := literal; // CodeCR470 JRL 4/5/13

frmMain.RPCBroker1.Param[7].Value := '0'; // Local mode vs. // CodeCR470 JRL 5/22/13

frmMain.RPCBroker1.Param[7].PType := literal; // Consolidated remote rpts // CodeCR470 JRL 5/22/13

try

DisplayBusyAnimation(True);

frmMain.RPCBroker1.Call;

except on E: Exception do

begin

ShowMessageCAPRI('DVBAB PENDING C&P REPORT returned an error!' + #13#10 + E.Message);

Screen.Cursor := crDefault;

exit;

end;

end;

panelReportChoice3.Visible := False;

ReportMemo.Visible := False;

If frmMain.RPCBroker1.Results.Count > 0 Then

Begin

if ADelimiter = '0' then // normal report // CodeCR470 JRL 4/5/13

begin

Repeat

Begin

Screen.Cursor := crHourglass;

If Pos('^', frmMain.RPCBroker1.Results[0]) > 0 Then

Begin

ReportMemo.Lines.Add(Copy(frmMain.RPCBroker1.Results[0], 1, Pos('^', frmMain.RPCBroker1.Results[0]) - 1));

frmMain.RPCBroker1.Results[0] := Copy(frmMain.RPCBroker1.Results[0], Pos('^', frmMain.RPCBroker1.Results[0]) + 1, Length(frmMain.RPCBroker1.Results[0]));

End

Else

Begin

ReportMemo.Lines.Add(frmMain.RPCBroker1.Results[0]);

frmMain.RPCBroker1.Results[0] := '';

End;

End

Until frmMain.RPCBroker1.Results[0] = '';

end

else // delimited report CodeCR470 JRL 4/5/13

begin

AbbrevReportName := 'PendingCPExam'; // CodeCR470 JRL 4/3/13

DelimitedLines := TStringList.Create; // CodeCR470 JRL 4/3/13

try

CAPRIQuickCopy(frmMain.RPCBroker1.Results, DelimitedLines);

FileName := SaveDelimitedReportCSV(DelimitedLines, AbbrevReportName

+ '\_'

+ FormatDateTime('YYYYMMDD', dtpInsuffStart.Date)

+ '\_'

+ FormatDateTime('YYYYMMDD', dtpInsuffEnd.Date));

ReportMemo.Lines.Insert(0, '');

if (FileName <> '') then

ReportMemo.Lines.Insert(0, IntToStr(DelimitedLines.Count)

+ ' Extract lines saved to: ' + #13#10

+ FileName)

else

ReportMemo.Lines.Insert(0, 'No delimited text lines were saved!');

finally

FreeAndNil(DelimitedLines);

end;

end;

Screen.Cursor := crDefault;

End;

If ReportMemo.Lines.count = 0 Then

ReportMemo.Lines.Add('No data found for requested report criteria.');

ReportMemo.Visible := True;

ReportMemo.SelStart := 0;

ReportMemo.SelLength := 0;

ReportMemo.SetFocus;

End;

panelReportChoice3.Visible := False;

End;

Table 48: Functions

| Function Name | GetRequestStatusReport | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified GUI and code to allow new Re-Routing statuses to be selected. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetRequestStatusReport | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

{==========================================================================

GetRequestStatusReport

Input: IsDelimited

IsIgnoreDate - ignore date range for NEW, PENDING, and RELEASED

This procedure retrieves the Request Status by Date report data.

//CodeCR266 - rpm 1/24/12

Added new statuses for re-routing - Patch 193 JRL 7/25/16

==========================================================================}

procedure Tfrmreports.GetRequestStatusReport(IsDelimited: Boolean; IsIgnoreDate: Boolean);

var

DelimitedLines: TStringList;

StatusFilter, DelimitedParam, IgnoreDateParam, FileName, AbbrevReportName: String;

RPCSuccess: Boolean;

begin

FileName := '';

AbbrevReportName := 'RequestStatusByDateReport';

case cbxRequestStatus.ItemIndex of

1: StatusFilter := 'N'; //NEW

2: StatusFilter := 'P'; //PENDING, REPORTED

3: StatusFilter := 'X'; //CANCELED BY MAS

4: StatusFilter := 'RX'; //CANCELED BY RO

5: StatusFilter := 'R'; //RELEASED TO RO, NOT PRINTED

6: StatusFilter := 'C'; //COMPLETED, PRINTED BY RO

7: StatusFilter := 'RN'; // Patch 193 JRL 7/25/16 // NEW, RE-ROUTED //

8: StatusFilter := 'RP'; // Patch 193 JRL 7/25/16 // RE-ROUTED, PENDING ACCEPTANCE //

9: StatusFilter := 'RA'; // Patch 193 JRL 7/26/16 // RE-ROUTED, ACCEPTED //

10: StatusFilter := 'RR'; // Patch 193 JRL 7/26/16 // RE-ROUTED, REJECTED //

else

StatusFilter := 'A'; //return all status values

end;

if IsIgnoreDate then

IgnoreDateParam := '1'

else

IgnoreDateParam := '0';

if IsDelimited then

DelimitedParam := '1'

else

DelimitedParam := '0';

DisplayBusyAnimation(TRUE);

RPCSuccess := CallRPC(frmMain.RPCBroker1, 'DVBAB REPORTS',

['14',

FMToDateConvert(FormatDateTime('mm/dd/yyyy', DateTimePicker1.DateTime))

+ '^'

+ FMToDateConvert(FormatDateTime('mm/dd/yyyy', DateTimePicker2.DateTime))

+ '^'

+ StatusFilter

+ '^'

+ DelimitedParam

+ '^'

+ IgnoreDateParam], nil, False);

DisplayBusyAnimation(FALSE);

if RPCSuccess and (frmMain.RPCBroker1.Results.Count > 0) then

begin

if NOT AnsiStartsText('NO DATA FOUND', frmMain.RPCBroker1.Results[0]) then

begin

if IsDelimited then

begin

DelimitedLines := TStringList.Create;

try

CapriQuickCopy(frmMain.RPCBroker1.Results, DelimitedLines);

if (DelimitedLines.Count > 0) then

begin

FileName := SaveDelimitedReportCSV(DelimitedLines, AbbrevReportName // CodeCR470 JRL 4/9/13

+ '\_' // Replaced TXT save with CSV save

+ FormatDateTime('YYYYMMDD', DateTimePicker1.Date)

+ '\_'

+ FormatDateTime('YYYYMMDD', DateTimePicker2.Date));

ReportMemo.Lines.Insert(0, '');

end;

if (FileName <> '') then

begin

ReportMemo.Lines.Insert(0, IntToStr(DelimitedLines.Count)

+ ' Extract lines saved to: ' + #13#10

+ FileName);

end

else

ReportMemo.Lines.Insert(0, 'No delimited text lines were saved!');

finally

FreeAndNil(DelimitedLines);

end;

end

else //regular report - not delimited

CapriQuickCopy(frmMain.RPCBroker1.Results, ReportMemo);

end

else //NO DATA FOUND

ReportMemo.Lines.Add('No data found for requested report criteria.');

end;

end;

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

{==========================================================================

GetRequestStatusReport

Input: IsDelimited

IsIgnoreDate - ignore date range for NEW, PENDING, and RELEASED

This procedure retrieves the Request Status by Date report data.

//CodeCR266 - rpm 1/24/12

Added new statuses for re-routing - Patch 193 JRL 7/25/16

==========================================================================}

procedure Tfrmreports.GetRequestStatusReport(IsDelimited: Boolean; IsIgnoreDate: Boolean);

var

DelimitedLines: TStringList;

StatusFilter, DelimitedParam, IgnoreDateParam, FileName, AbbrevReportName: String;

RPCSuccess: Boolean;

begin

FileName := '';

AbbrevReportName := 'RequestStatusByDateReport';

case cbxRequestStatus.ItemIndex of

1: StatusFilter := 'N'; //NEW

2: StatusFilter := 'P'; //PENDING, REPORTED

3: StatusFilter := 'X'; //CANCELED BY MAS

4: StatusFilter := 'RX'; //CANCELED BY RO

5: StatusFilter := 'R'; //RELEASED TO RO, NOT PRINTED

6: StatusFilter := 'C'; //COMPLETED, PRINTED BY RO

7: StatusFilter := 'RN'; // Patch 193 JRL 7/25/16 // NEW, RE-ROUTED //

8: StatusFilter := 'RP'; // Patch 193 JRL 7/25/16 // RE-ROUTED, PENDING ACCEPTANCE //

9: StatusFilter := 'RA'; // Patch 193 JRL 7/26/16 // RE-ROUTED, ACCEPTED //

10: StatusFilter := 'RR'; // Patch 193 JRL 7/26/16 // RE-ROUTED, REJECTED //

else

StatusFilter := 'A'; //return all status values

end;

if IsIgnoreDate then

IgnoreDateParam := '1'

else

IgnoreDateParam := '0';

if IsDelimited then

DelimitedParam := '1'

else

DelimitedParam := '0';

DisplayBusyAnimation(TRUE);

RPCSuccess := CallRPC(frmMain.RPCBroker1, 'DVBAB REPORTS',

['14',

FMToDateConvert(FormatDateTime('mm/dd/yyyy', DateTimePicker1.DateTime))

+ '^'

+ FMToDateConvert(FormatDateTime('mm/dd/yyyy', DateTimePicker2.DateTime))

+ '^'

+ StatusFilter

+ '^'

+ DelimitedParam

+ '^'

+ IgnoreDateParam], nil, False);

DisplayBusyAnimation(FALSE);

if RPCSuccess and (frmMain.RPCBroker1.Results.Count > 0) then

begin

if NOT AnsiStartsText('NO DATA FOUND', frmMain.RPCBroker1.Results[0]) then

begin

if IsDelimited then

begin

DelimitedLines := TStringList.Create;

try

CapriQuickCopy(frmMain.RPCBroker1.Results, DelimitedLines);

if (DelimitedLines.Count > 0) then

begin

FileName := SaveDelimitedReportCSV(DelimitedLines, AbbrevReportName // CodeCR470 JRL 4/9/13

+ '\_' // Replaced TXT save with CSV save

+ FormatDateTime('YYYYMMDD', DateTimePicker1.Date)

+ '\_'

+ FormatDateTime('YYYYMMDD', DateTimePicker2.Date));

ReportMemo.Lines.Insert(0, '');

end;

if (FileName <> '') then

begin

ReportMemo.Lines.Insert(0, IntToStr(DelimitedLines.Count)

+ ' Extract lines saved to: ' + #13#10

+ FileName);

end

else

ReportMemo.Lines.Insert(0, 'No delimited text lines were saved!');

finally

FreeAndNil(DelimitedLines);

end;

end

else //regular report - not delimited

CapriQuickCopy(frmMain.RPCBroker1.Results, ReportMemo);

end

else //NO DATA FOUND

ReportMemo.Lines.Add('No data found for requested report criteria.');

end;

end;

##### GUI – ReRouteReject.pas

Table 36: GUI

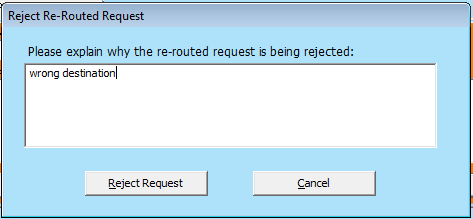
| Unit Name | Description |
| --- | --- |
| ReRouteReject.pas | Add form to allow user to enter a description why they are rejecting a re-routed request. |

##### GUI Classes

##### Current Form

New form.

##### Modified Form



##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| memoReRouteRejectReason: | TMemo; | Reject Reason memo box |
| lblReRouteReject: | TLabel; | Reject Re-route label |
| btnOK: | TButton; | Ok button |
| btnCancelReject: | TButton; | Cancel button |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| btnOKClick | Button Click | Ok button clicked, send Reject Info. |
| btnCancelRejectClick | Button Click | Cancel button clicked, return to View Exam screen. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| SendRejectionInfo | Function | Send Rejection information. |

##### Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
| CLOSE\_FORM = '1'; | Constant | Form Close flag |
| DONT\_CLOSE\_FORM = '2'; | Constant | Form Close flag |
| OriginalIEN : String; | Form Variable | Value saved for Rejection email message |

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | RerouteReject | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** | Enter rejection reason when rejecting a re-routed request. | | | |

| Current Form Layout |
| --- |
| N/A |

| Modified Form Layout (Changes are in bold) |
| --- |
| See modified form above |

##### Functions

Table 48: Functions

| Function Name | btnOKClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Ok button clicked, send Reject Info. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | n/a | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnOKClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// OK Button Clicked

// -----------------------------------------------------------------------------

// Database restrictions:

// Minimum length: 3 characters

// Maximum length: 200 characters

// ReturnValue is not being used to close or not close the ViewExam screen.

// -----------------------------------------------------------------------------

procedure TfrmReRouteReject.btnOKClick(Sender: TObject);

const

SUCCESS = TRUE;

FAILURE = FALSE;

var

Unused : Boolean;

begin

if length(memoReRouteRejectReason.text) >= 3 then // database requires 3 or more characters

begin

Unused := SendRejectionInfo; // return value is not being use, always close form

frmViewExam.ReturnValue := CLOSE\_FORM;

Close;

end

else

begin

ShowMessageCAPRI('You must enter a full explanation for rejecting this request.');

memoReRouteRejectReason.Lines.Clear;

memoReRouteRejectReason.SetFocus;

end;

end;

Table 48: Functions

| Function Name | btnCancelRejectClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Cancel button clicked, return to View Exam screen. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnCancelRejectClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Cancel Button Clicked

// -----------------------------------------------------------------------------

procedure TfrmReRouteReject.btnCancelRejectClick(Sender: TObject);

begin

frmViewExam.ReturnValue := DONT\_CLOSE\_FORM;

Close;

end;

Table 48: Functions

| Function Name | SendRejectionInfo | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Send Rejection information. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | SendRejectionInfo | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// Send Rejection Info

// -----------------------------------------------------------------------------

// This will be a two-step process. First, we reject the request and the

// status information is updated on the local VistA site. Successful rejecting

// at the local site provides the station number and original IEN of the

// originating site. We call the same RPC again via the XWB DIRECT RPC call

// to update the originating site.

//

// RPC Input:

// REQUEST IEN

// ACCEPT 'A' or REJECT 'R'

// REJECT REASON (255 CHARACTERS)

// RPC Output (Reports sending of MailMan message):

// 1: SUCCESS = 1 or FAILURE = 0 ^

// 2: SUCCESS or FAILURE Message ^

// The following two values are only returned when we are updating the site

// we are logged into. If we are calling via the XWB DIRECT RPC call, only

// the first two pieces are returned.

// 3: Station Number ^ (for notifying originating site) - if no station

// number is returned, then originating site is the same

// 4: Original IEN (for notifying originating site)

// -----------------------------------------------------------------------------

function TfrmReRouteReject.SendRejectionInfo : Boolean;

CONST

RPC\_DIRECT = 'XWB DIRECT RPC';

RPCNAME = 'DVBA CAPRI REROUTE STATUS';

REJECT = 'R';

SUCCESS = '1';

FAILURE = '0';

var

Status, StatusMessage, StationNum, buffer : string;

begin

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

Result := FALSE;

// Update local VistA with REJECT status

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].Value := frmViewExam.edExamReferenceNumber.Text; // Request IEN

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[1].Value := REJECT;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[2].Value := memoReRouteRejectReason.Text;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except // RPC Error - RPC failed, user exits

On E: EBrokerError Do

begin

ShowMessageCAPRI(RPCNAME + ' could not be accessed! '+#10#13+E.Message);

Exit; // allow user to try again or hit cancel

end;

End; // try

If frmMain.RPCBroker1.Results.Count > 0 Then

begin // RPC returned results - process them

Status := Piece(frmMain.RPCBroker1.Results[0], '^', 1);

StatusMessage := Piece(frmMain.RPCBroker1.Results[0], '^', 2);

StationNum := Piece(frmMain.RPCBroker1.Results[0], '^', 3);

OriginalIEN := Piece(frmMain.RPCBroker1.Results[0], '^', 4);

if Status = SUCCESS then // Success, request was rejected

begin

// Local VistA site was updated successfully.

Application.ProcessMessages;

// Local Re-Routes do not need to update the original site. All status

// updates will happen on the first RPC call for accepting this request

if not frmViewExam.LocalReRoute then

begin

// Update Originating site's status (same RPC called via WXB DIRECT RPC)

frmMain.RPCBroker1.RemoteProcedure := RPC\_DIRECT;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := StationNum;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := ''; // RPC version parameter (Optional)

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := OriginalIEN;

frmMain.RPCBroker1.Param[4].Value := REJECT;

frmMain.RPCBroker1.Param[4].PType := literal;

frmMain.RPCBroker1.Param[5].Value := memoReRouteRejectReason.Text;

frmMain.RPCBroker1.Param[5].PType := literal;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

begin // RPC returned results - process them

Status := Piece(frmMain.RPCBroker1.Results[0], '^', 1);

StatusMessage := Piece(frmMain.RPCBroker1.Results[0], '^', 2); // debug only, not displaying

// FYI: An email is sent to a mail group (DVBA C 2507 REROUTE) at the originating site indicating a rejection

if Status = SUCCESS then

Result := TRUE

else // if Status = FAILURE then // Originating VistA site was not updated

begin

buffer := RPC\_DIRECT + ' : ' + RPCNAME + #13#10 +

' did not update the 2507 Request Status correctly. ' +

frmViewExam.edtReRoutedFrom.Text + ' has not been notified.';

ShowMessageCAPRI(buffer);

end;

end; // if RPCBroker1.Results.Count > 0

except

On E: EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Station Number ' + StationNum + '!'+#13#10+E.Message);

end;

end; // try except

end;

end // Status = SUCCESS

else // Status = FAILURE

begin

buffer := RPCNAME + ' did not update the 2507 Request Status correctly.' +

frmViewExam.edtReRoutedFrom.Text + ' has not been notified.';

ShowMessageCAPRI(buffer);

end;

end // RPCBroker1.Results.Count > 0

else // RPC call returned no results

begin

buffer := RPCNAME + ' failed. ' + frmViewExam.edtReRoutedFrom.Text + ' has not been notified.';

ShowMessageCAPRI(buffer);

end;

end;

##### GUI – edtRerouteReasons.pas

Table 36: GUI

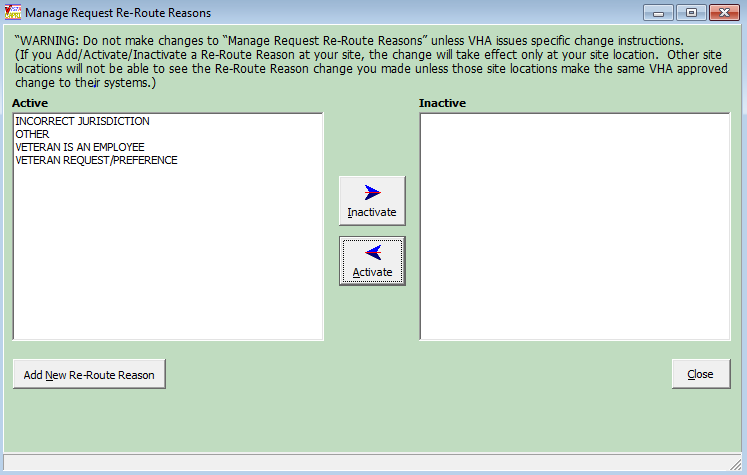
| Unit Name | Description |
| --- | --- |
| edtRerouteReasons.pas | Allow Re-Route Reasons to be activated, inactivated and added. A security key is needed to see this screen. |

##### GUI Classes

##### Current Form

New form.

##### Modified Form



##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| Panel1 | TPanel | Back panel |
| StatusBar1 | TStatusBar | Bottom status bar |
| txtActive | TStaticText | Active label |
| lbActive | TListBox | Active listbox |
| txtInactive | TStaticText | Inactive label |
| lbInactive | TListBox | Inactive Listbox |
| btnMoveToInactive | TBitBtn | Move to Inactive button |
| btnMoveToActive | TBitBtn | Move to Active button |
| btnAddReason | TBitBtn | Add Reason button |
| txtAddName | TStaticText | Add label |
| FMEditName | TFMEdit | Edit name label |
| BtnSaveNewActive | TBitBtn | Save button |
| btnCancel | TBitBtn | Cancel button |
| btnClose | TButton | Close button |
| txtStatus | TStaticText | Status label |
| FMEditStatus | TFMEdit | Status edit box |
| btnGetList | TButton | Get List button |
| btnDeleteActiveReason | TButton | Delete Active Reason button |
| btnDeleteInactive | TButton | Delete Inactive Reason button |
| FMLister1 | TFMLister | FMLister |
| FMGets1 | TFMGets | FMGets |
| FMFiler1 | TFMFiler | FMFiler |
| VA508StaticText1 | TVA508StaticText | Static Text |
| Memo1 | TMemo | Memo box |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| FormCreate | Form Create | Added form to allow the user to Add/Inactivate Re-Routing Reasons. Menu option is governed by a security key and is not available to VBA. |
| btnCloseClick | Button Click | Close form. |
| btnAddReasonClick | Button Click | Add Re-Route Reason button clicked. |
| btnDeleteInactiveClick | Button Click | Delete Inactive Re-Route Reason button. Debug only. Not used. |
| btnMoveToInactiveClick | Button Click | Inactivate Re-Route Reason button clicked. |
| btnMoveToActiveClick | Button Click | Activate Re-Route Reason button clicked. |
| FormShow | Form Show | Form setup routine. |
| btnSaveNewActiveClick | Button Click | Save new Re-Route Reason button clicked. |
| btnCancelClick | Button Click | Cancel new Re-Route Reason button clicked. |
| btnGetListClick | Button Click | Get List. Debug routine. Not used. |
| FormDestroy | Form Destroy | Form close down routine. |
| btnDeleteActiveReasonClick | Button Click | Delete Active Re-Route Reason button. Debug only. Not used. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| DeleteReason | Function | Delete Re-Route Reason. Debug routine. Not Used. |
| LoadDataIntoListboxes | Procedure | Load listbox data. |
| FreeListBoxes | Procedure | Clean up memory. |
| FileChanges | Function | Update all changes. |
| UpdateItemState | Function | Update the active/inactive state of the selected item. |
| SaveNewItem | Function | Save new item |
| SaveItem | Function | Save Items. |
| IsDelphiRunning | Procedure | Debug Routine. Not used. |

##### Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
| TIENInfo | Class | Class of IEN strings |
| AddRecord | Form Variable | Flag for adding record |
| AddedIENS | Form Variable | IEN Added value |

##### Class Events – N/A in Requirement 2.6.1

##### Class Methods – N/A in Requirement 2.6.1

##### Class Properties – N/A in Requirement 2.6.1

##### Uses Clause

|  |  |
| --- | --- |
| Interface | Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, Fmcmpnts, Diaccess, StdCtrls, Fmcntrls, Buttons, ComCtrls, ExtCtrls, VA508AccessibilityManager; |
| Implementation | Main, DiTypLib, VA508AccessibilityRouter; |

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | edtRerouteReasons | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** | Allow user to add/inactivate Re-Route Reasons | | | |

| Current Form Layout |
| --- |
| New form. |

| Modified Form Layout (Changes are in bold) |
| --- |
| See Modified Form above. |

##### Functions

Table 48: Functions

| Function Name | FormCreate | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Added form to allow the user to Add/Inactivate Re-Routing Reasons. Menu option is governed by a security key and is not available to VBA. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormCreate | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Form Create

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.FormCreate(Sender: TObject);

begin

frmMain.RPCBroker1.Connected := True;

btnAddReason.Visible := TRUE;

FMEditName.Hide;

FMEditStatus.Hide;

btnSaveNewActive.Visible := FALSE;

btnCancel.Visible := FALSE;

txtAddName.Visible := FALSE;

txtStatus.Hide;

end;

Table 48: Functions

| Function Name | btnCloseClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Close form. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnCloseClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Close button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnCloseClick(Sender: TObject);

begin

Close;

end;

Table 48: Functions

| Function Name | btnAddReasonClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Add Re-Route Reason button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnAddReasonClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Add Reason button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnAddReasonClick(Sender: TObject);

begin

StatusBar1.SimpleText := '';

btnAddReason.Visible := FALSE;

txtAddName.Visible := TRUE;

FMEditName.Show;

btnSaveNewActive.Visible := TRUE;

btnCancel.Visible := TRUE;

FMEditName.Enabled := True;

FMEditName.FMRequired := False;

FMEditName.FMCtrlOptions := [];

FMEditStatus.Enabled := False;

FMEditStatus.FMRequired := False;

FMEditStatus.FMCtrlOptions := [];

FMFiler1.Clear;

FMGets1.ClearCtrlData;

FMGets1.IENS := '+1,';

FMGets1.StuffIENS('+1,');

FMEditName.FMRequired := True;

// FMEditName.FMCtrlOptions := [coValOnExit];

FMEditName.Text := 'Enter Reason';

FMEditName.FMCtrlInternal := 'ENTER REASON';

FMEditStatus.FMCtrlOptions := [coValOnExit];

FMEditStatus.Text := 'A';

FMEditStatus.FMCtrlInternal := '1'; // Status = Active

if FMEditName.CanFocus then

FMEditName.SetFocus;

AddRecord := True;

end;

Table 48: Functions

| Function Name | btnDeleteInactiveClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Delete Inactive Re-Route Reason button. Debug only. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnDeleteInactiveClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// DEBUG ROUTINE - Delete Inactive Reason button Clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnDeleteInactiveClick(Sender: TObject);

var

IENS: String;

tmpobj: TObject;

begin

StatusBar1.SimpleText := '';

if lbActive.ItemIndex >= 0 then begin

tmpobj := lbActive.Items.Objects[lbActive.ItemIndex];

if tmpobj <> nil then begin

IENS := TIENInfo(tmpobj).IEN;

DeleteReason(IENS);

end;

end;

end;

Table 48: Functions

| Function Name | btnMoveToInactiveClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Inactivate Re-Route Reason button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnMoveToInactiveClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Move to Inactive button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnMoveToInactiveClick(Sender: TObject);

var

Index, idx: Integer;

IENS: String;

tmpobj: TObject;

AIENInfo: TIENInfo;

begin

StatusBar1.SimpleText := '';

Index := lbActive.ItemIndex;

if Index >= 0 then begin

tmpobj := lbActive.Items.Objects[Index];

if tmpobj <> nil then begin

IENS := TIENInfo(tmpobj).IEN;

if UpdateItemState(IENS, lbActive.Items[Index], FALSE) then begin

AIENInfo := TIENInfo.Create;

try

AIENInfo.IEN := IENS;

idx := lbInactive.Items.AddObject(lbActive.Items[Index], AIENInfo);

finally

tmpobj.Free;

lbActive.Items.Delete(Index);

end;

end;

end;

end;

end;

Table 48: Functions

| Function Name | btnMoveToActiveClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Activate Re-Route Reason button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnMoveToActiveClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Move to Active button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnMoveToActiveClick(Sender: TObject);

var

Index, idx: Integer;

IENS: String;

tmpobj: TObject;

AIENInfo: TIENInfo;

begin

StatusBar1.SimpleText := '';

Index := lbInactive.ItemIndex;

if Index >= 0 then begin

tmpobj := lbInactive.Items.Objects[Index];

if tmpobj <> nil then begin

IENS := TIENInfo(tmpobj).IEN;

if UpdateItemState(IENS, lbInactive.items[Index], TRUE) then begin

AIENInfo := TIENInfo.Create;

try

AIENInfo.IEN := IENS;

idx := lbActive.Items.AddObject(lbInactive.Items[Index], AIENInfo);

finally

tmpobj.Free;

lbInactive.Items.Delete(Index);

end;

end;

end;

end;

end;

Table 48: Functions

| Function Name | FormShow | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Form setup routine. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormShow | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Form Show

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.FormShow(Sender: TObject);

begin

LoadDataIntoListBoxes;

IsDelphiRunning; // Show test buttons if Delphi is running

end;

Table 48: Functions

| Function Name | btnSaveNewActiveClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Save new Re-Route Reason button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnSaveNewActiveClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Save New Active Reason

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnSaveNewActiveClick(Sender: TObject);

var

idx: Integer;

bret: Boolean;

AIENInfo: TIENInfo;

begin

bret := SaveNewItem;

if bret then begin

AIENInfo := TIENInfo.Create;

try

AIENInfo.IEN := FMGets1.IENS;

if FMEditName.Text > '' then

idx := lbActive.Items.AddObject(FMEditName.Text, AIENInfo);

finally

txtAddName.Visible := FALSE;

btnAddReason.Visible := TRUE;

FMEditName.Hide;

txtStatus.Hide;

FMEditStatus.Hide;

btnSaveNewActive.Visible := FALSE;

btnCancel.Visible := FALSE;

end;

end;

end;

Table 48: Functions

| Function Name | btnCancelClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Cancel new Re-Route Reason button clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnCancelClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Cancel button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnCancelClick(Sender: TObject);

begin

StatusBar1.SimpleText := '';

txtAddName.Visible := FALSE;

FMEditName.Hide;

txtStatus.Hide;

FMEditStatus.Hide;

btnSaveNewActive.Visible := FALSE;

btnCancel.Visible := FALSE;

btnAddReason.Visible := TRUE;

FMFiler1.Clear;

end;

Table 48: Functions

| Function Name | btnGetListClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Get List. Debug routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnGetListClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// DEBUG ROUTINE - Get List button clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnGetListClick(Sender: TObject);

begin

LoadDataIntoListBoxes;

end;

Table 48: Functions

| Function Name | FormDestroy | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Form close down routine. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormDestroy | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Form Destroy

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.FormDestroy(Sender: TObject);

begin

FreeListBoxes;

end;

Table 48: Functions

| Function Name | btnDeleteActiveReasonClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Delete Active Re-Route Reason button. Debug only. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnDeleteActiveReasonClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// DEBUG ROUTINE - Delete Active Reason button Clicked

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.btnDeleteActiveReasonClick(Sender: TObject);

var

IENS: String;

tmpobj: TObject;

begin

StatusBar1.SimpleText := '';

if lbActive.ItemIndex >= 0 then begin

tmpobj := lbActive.Items.Objects[lbActive.ItemIndex];

if tmpobj <> nil then begin

IENS := TIENInfo(tmpobj).IEN;

DeleteReason(IENS);

end;

end;

end;

Table 48: Functions

| Function Name | DeleteReason | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Delete Re-Route Reason. Debug routine. Not Used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | DeleteReason | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// DEBUG ROUTINE - Delete Reason

//------------------------------------------------------------------------------

function TfrmEdtReRouteReasons.DeleteReason(inIENS: String): Boolean;

begin

Result := False;

StatusBar1.SimpleText := '';

FMGets1.IENS := inIENS;

if DeleteRecord(frmMain.RPCBroker1, FMGets1.FileNumber, FMGets1.IENS) then begin

LoadDataIntoListBoxes;

Result := True;

end

else

StatusBar1.SimpleText := 'There was a problem deleting the record.';

end;

Table 48: Functions

| Function Name | LoadDataIntoListboxes | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Load listbox data. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | LoadDataIntoListboxes | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Load Data Into List Boxes

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.LoadDataIntoListboxes;

var

i, icnt, idx: Integer;

fmRecObj: TFMRecordObj;

name, status, IENS: String;

AIENInfo: TIENInfo;

begin

// Initialize listboxes

FreeListBoxes;

FMLister1.GetList(nil);

if FMLister1.ErrorList.Count > 0 then

FMLister1.DisplayErrors

else begin

icnt := FMLister1.Results.Count;

for I := 0 to icnt - 1 do begin

IENS := FMLister1.Results[i];

if IENS > '' then begin

fmRecObj := TFMRecordObj(FMLister1.Results.Objects[i]);

name := fmRecObj.GetField('.01').FMDBExternal;

status := uppercase(fmRecObj.GetField('.02').FMDBExternal);

AIENInfo := TIENInfo.Create;

AIENInfo.IEN := IENS;

if status = 'ACTIVE' then begin // inactive = no

idx := lbActive.Items.AddObject(name, AIENInfo);

end

else begin // inactive = yes

idx := lbInactive.Items.AddObject(name, AIENInfo);

end;

end;

end;

end;

end;

Table 48: Functions

| Function Name | FreeListBoxes | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Clean up memory. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FreeListBoxes | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Free List Boxes

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.FreeListBoxes;

var

i: Integer;

AIENInfo: TIENInfo;

begin

for I := 0 to lbActive.Count - 1 do

if lbActive.Items.Objects[i] <> nil then begin

AIENInfo := TIENInfo(lbActive.Items.Objects[i]);

AIENInfo.Free;

end;

lbActive.Clear;

for I := 0 to lbInactive.Count - 1 do

if lbInactive.Items.Objects[i] <> nil then begin

AIENInfo := TIENInfo(lbInactive.Items.Objects[i]);

AIENInfo.Free;

end;

lbInactive.Clear;

end;

Table 48: Functions

| Function Name | FileChanges | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Update all changes. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FileChanges | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// File Changes

//------------------------------------------------------------------------------

function TfrmEdtReRouteReasons.FileChanges: Boolean;

begin

Result := False;

StatusBar1.SimpleText := '';

if FMFiler1.DataProblemCheck then

FMFiler1.ProcessDataProblemList

else if FMFiler1.Update then begin // changes filed

Result := True;

end

else

FMFiler1.DisplayErrors;

end;

Table 48: Functions

| Function Name | UpdateItemState | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Update the active/inactive state of the selected item. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | UpdateItemState | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Update Item State

//------------------------------------------------------------------------------

function TfrmEdtReRouteReasons.UpdateItemState(ItemIEN, ItemName: string;

ItemState: Boolean): Boolean;

begin

// Update Item State

if AddRecord then

btnCancel.Click;

AddRecord := False;

Result := SaveItem(ItemIEN, ItemName, ItemState);

end;

Table 48: Functions

| Function Name | SaveNewItem | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Save new item | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | SaveNewItem | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

function TfrmEdtReRouteReasons.SaveNewItem: Boolean;

begin

// Save Item and return save status

Result := SaveItem('', '', True);

end;

Table 48: Functions

| Function Name | SaveItem | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Save Items. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | SaveItem | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Save Item

//------------------------------------------------------------------------------

function TfrmEdtReRouteReasons.SaveItem(ItemIEN, ItemName: String;

ItemState: Boolean): Boolean;

var

update: Boolean;

IENS: String;

cpos, ilen: Integer;

procedure StatusMsg(inMsg: String);

begin

StatusBar1.SimpleText := inmsg;

GetScreenReader.Speak(inmsg);

end;

begin

StatusBar1.SimpleText := '';

IENS := '';

update := False;

Result := True;

if ItemIEN > '' then

FMGets1.IENS := ItemIEN;

if ItemName > '' then begin

FMEditName.Text := ItemName;

FMEditName.FMCtrlInternal := ItemName;

end

else

FMEditName.FMCtrlInternal := FMEditName.Text;

if ItemState then begin

FMEditStatus.Text := 'A';

FMEditStatus.FMCtrlInternal := '1'; // Status = active

end

else begin

FMEditStatus.Text := 'I';

FMEditStatus.FMCtrlInternal := '0'; // Status = inactive

end;

if (FMGets1.IENS > '') then begin

StatusBar1.SimpleText := '';

ilen := Length(FMEditname.Text);

if ilen < 3 then begin

StatusMsg('Reason must be at least three characters in length.');

if FMEditName.CanFocus then

FMEditName.SetFocus;

Result := False;

exit;

end;

FMEditName.IENS := FMGets1.IENS;

if FMEditName.FMModified then

if FMEditName.Validate then begin

FMFiler1.AddChgdControl(FMEditName);

FMEditName.Text := FMEditName.FMCtrlExternal;

end

else begin

frmMain.FMValidator1.DisplayErrors;

FMEditName.Text := FMEditName.FMTag;

Result := False;

end;

FMEditStatus.IENS := FMGets1.IENS;

if FMEditStatus.FMModified then

if FMEditStatus.Validate then begin

FMFiler1.AddChgdControl(FMEditStatus);

FMEditStatus.Text := FMEditStatus.FMCtrlExternal;

end

else begin

frmMain.FMValidator1.DisplayErrors;

FMEditStatus.Text := FMEditStatus.FMTag;

Result := False;

end;

if Result then begin

if FMFiler1.AnythingToFile then begin

update := FileChanges;

Result := update;

if update then begin

if AddRecord then begin

IENS := FMFiler1.FindIen('+1,') + ',';

if IENS <> ',' then begin

cpos := Pos(',', IENS);

AddedIENS := Copy(IENS, 1, cpos - 1);

StatusMsg('Record added.');

Result := True;

end

else begin

IENS := '';

StatusMsg('New record not found.');

Result := False;

end;

end; // if AddRecord

if IENS > '' then begin

FMGets1.IENS := IENS;

FMGets1.GetAndFill;

end; // if IENS > ''

end; // if update

end // if AnythingToFile

else

StatusMsg('No changes to file!');

end; // if Result

end; // if IENS > ''

AddRecord := False;

AddedIENS := '';

Invalidate;

end;

Table 48: Functions

| Function Name | IsDelphiRunning | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | IsDelphiRunning | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Is Delphi Running?

//------------------------------------------------------------------------------

// If Delphi is running, we are in the IDE and operating in a testing mode.

// Display testing buttons. If Delphi is not running, do not display test

// buttons.

//------------------------------------------------------------------------------

// Debug routines were included because they existed in the Cancellation

// Reasons code and this form was derived from that. I've added the

// IsDelphiRunning code to show the test buttons in the development mode but

// none of the test routines have been "tested" for this form.

//------------------------------------------------------------------------------

procedure TfrmEdtReRouteReasons.IsDelphiRunning;

begin

if DebugHook <> 0 then

begin

btnGetList.Visible := TRUE;

btnGetList.Enabled := TRUE;

btnDeleteActiveReason.Visible := TRUE;

btnDeleteActiveReason.Enabled := TRUE;

btnDeleteInactive.Visible := TRUE;

btnDeleteInactive.Enabled := TRUE;

end

else

begin

btnGetList.Visible := FALSE;

btnGetList.Enabled := FALSE;

btnDeleteActiveReason.Visible := FALSE;

btnDeleteActiveReason.Enabled := FALSE;

btnDeleteInactive.Visible := FALSE;

btnDeleteInactive.Enabled := FALSE;

end;

end;

##### GUI – ReRouteRequest.pas

Table 36: GUI

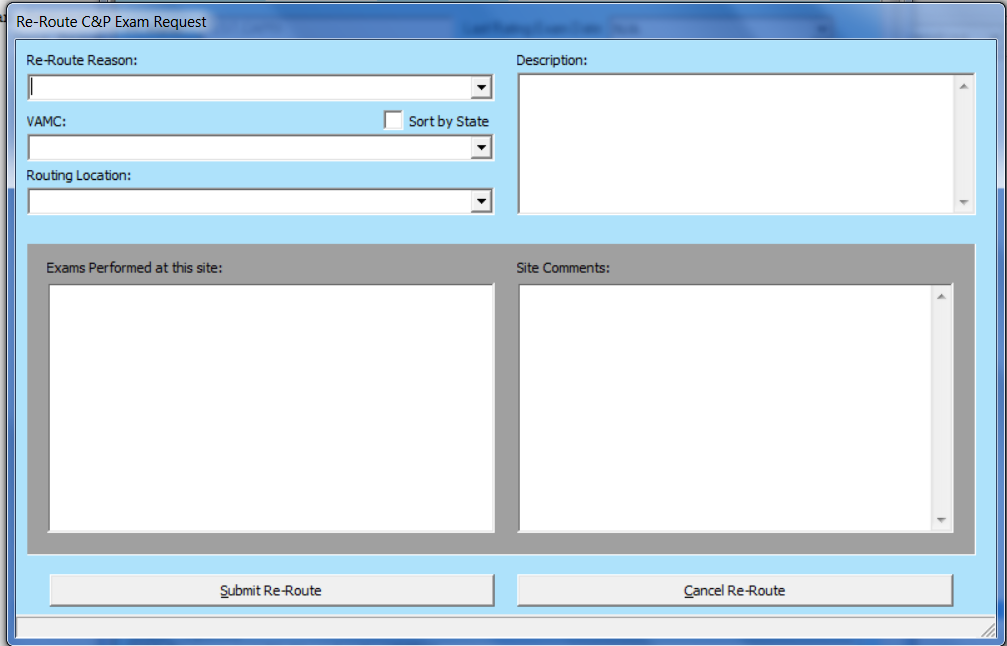
| Unit Name | Description |
| --- | --- |
| ReRouteRequest.pas | This is a new form used for defining parameters for re-routing a request. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form

New form.

##### Modified Form



##### Components on Form

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| Panel1 | TPanel | Back panel |
| FMReRouteReason | TFMComboBox | Re-Route Reason combobox |
| FMReRouteDescriptoin | TFMMemo | Re-Route Description memo box |
| FMReRouteLister | TFMLister | FMLister |
| VA508AccessibilityManager1 | TVA508AccessibilityManager | VA508 |
| Panel2 | TPanel | Back panel |
| btnCancelReRoute | TButton | Cancel button |
| btnRerouteRequest | TButton | Re-Route Request button |
| CbxLocation | TCombobox | Routing Location combobox |
| sttxtReRouteReason | TStaticText | Re-Route Reason label |
| sttxtReRouteDescription | TStaticText | Re-Route Description label |
| sttxtReRouteLocation | TStaticText | Re-Route Location label |
| sttxtVAMC | TStaticText | VAMC label |
| FMVAMC | TFMComboBox | VAMC combobox |
| sttxtSortByState | TStaticText | Sort by State label |
| cbxSortByState | TCheckBox | Sort by State checkbox |
| Panel3 | TPanel | Back panel |
| FMExamList | TFMListBox | Exam Listbox |
| sttxtExams | TStaticText | Exam label |
| sttxtSiteComments | TStaticText | Site Comments label |
| FMDivisionComments | TFMMemo | Site Comments memo box |
| btnTestGuestLogin | TButton | Debug button |
| btnTestRemoteRPC | TButton | Debug button |
| StatusBar1 | TStatusBar | Status bar |
| btnTestRoutingLocation | TButton | Debug button |
| btnTestExamList | TButton | Debug button |
| btnTestSiteComments | TButton | Debug button |
| edtStationNumber | TEdit | Debug edit |

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| btnCancelReRouteClick | Button Click | Cancel Re-Route button clicked. Close form. |
| btnRerouteRequestClick | Button Click | Re-Route Request button clicked. Sent re-route information. |
| btnTestGuestLoginClick | Button Click | Debug Routine. Not used. |
| FMVAMCExit | Combobox exit | Selected VAMC. Populate Routing locations drop-down. |
| cbxLocationChange | Combobox change | Selected Routing Location. Populated exam list and site comments. |
| FormClose | Form Close | Close the form. |
| FormCreate | Form Create | Setup the form. |
| btnTestRemoteRPCClick | Button Click | Debug Routine. Not used. |
| btnTestRoutingLocationsClick | Button Click | Debug Routine. Not used. |
| btnTestExamListClick | Button Click | Debug Routine. Not used. |
| btnTestSiteCommentsClick | Button Click | Debug Routine. Not used. |
| cbxSortByStateClick | Combobox change | Sort by state checkbox clicked. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| CreateReRouteRequest | Procedure | Setup parameters and call RPC to create a re-routed request. |
| CheckUserInputValues | Function | Check input values to make certain all values are completed before re-routing a request. |
| GetRoutingLocations | Procedure | Retrieve the Routing Locations from the remote VistA instance selected in the VAMC list. |
| GetVAMCLocations | Procedure | Retrieve the VAMC list. |
| GetExamLists | Procedure | Retrieve the exam lists from the remote VistA instance selected in the VAMC list. |
| GetSiteComments | Procedure | Retrieve the site comments from the remote VistA instance selected in the VAMC list. |
| GetReRouteReasons | Procedure | Retrieve the Re-Route Reasons and populate the drop-down list. |
| IsDelphiRunning | Procedure | Debug Routine. Not used. |
| CheckIfVAMCCanHandleReroutes | Function | Check if the VAMC selected has Patch 193 installed before allowing a re-route. |

##### Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
| CLOSE\_FORM = '1'; | Form Constant | Indicator value for closing form |
| DONT\_CLOSE\_FORM = '2'; | Form Constant | Indicator value for closing form |
| RoutingLocation\_IEN : String; | Private Form Variable | Selected Routing Location IEN |
| VAMC\_IEN : String; | Private Form Variable | Selected VAMC IEN |
| VAMC\_IEN\_List : TStringList; | Private Form Variable | List of VAMC IENs |
| VAMC\_Station\_List : TStringList; | Private Form Variable | List of VAMC Station Numbers |
| VAMC\_Name\_List : TStringList; | Private Form Variable | List of VAMC Names |
| VAMC\_State\_List : TStringList; | Private Form Variable | List of VAMC by State Names |
| RoutingLocation\_IEN\_List : TStringList; | Private Form Variable | List of Routing Locations IENs |
| ReRouteReasonIENList : TStringList; | Private Form Variable | List of Re-Route Reason IENs |
| SelectedStation : String; | Private Form Variable | Selected Station name |

##### Class Events – Not Applicable for Requirement 2.6.1

##### Class Methods – Not Applicable for Requirement 2.6.1

##### Class Properties – Not Applicable for Requirement 2.6.1

##### Uses Clause

|  |  |
| --- | --- |
| Interface | Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Fmcntrls, VA508AccessibilityManager, ORCtrls, ExtCtrls, CAPRISupport, dialogsCAPRI, Diaccess, Fmcmpnts, Trpcb, untMiscMthds, CCOWRPCBroker, CCOWRPCBrokerCAPRI, ComCtrls; |
| Implementation | Main, rofinder, unitzip, viewexam, essoselect; |

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** | RerouteRequest | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** | Re-Route a 2507 C&P Request | | | |

| Current Form Layout |
| --- |
| New form |

| Modified Form Layout (Changes are in bold) |
| --- |
| See form above |

##### Functions

Table 48: Functions

| Function Name | btnCancelReRouteClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Cancel Re-Route button clicked. Close form. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnCancelReRouteClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Cancel Re-Route Button Clicked

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnCancelReRouteClick(Sender: TObject);

begin

// No action needed except to close the form

frmViewExam.ReturnValue := DONT\_CLOSE\_FORM;

Close;

end;

Table 48: Functions

| Function Name | btnRerouteRequestClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Re-Route Request button clicked. Sent re-route information. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnRerouteRequestClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Re-Route Request button clicked

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnRerouteRequestClick(Sender: TObject);

var

request\_rerouted : Boolean;

CloseForm : Boolean;

begin

if CheckUserInputValues then // if all inputs are satisfied, allow re-routing

begin

StatusBar1.SimpleText := 'Re-Routing Request';

request\_rerouted := CreateReRouteRequest; // send the re-routed request

if request\_rerouted = TRUE then // re-route was successful

frmViewExam.ReturnValue := CLOSE\_FORM

else

frmViewExam.ReturnValue := DONT\_CLOSE\_FORM;

Close;

end; // CheckUserInputValues

end;

Table 48: Functions

| Function Name | btnTestGuestLoginClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnTestGuestLoginClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// TEST ROUTINE - TEST GUEST LOGIN TO TEST CLAIMS SERVER

//------------------------------------------------------------------------------

// This routine logs a "local user" silently onto the TEST CLAIMS system and

// retrieves data via a Fileman control as well as via a RPC. Proof of concept

// for logging in silently to TEST CLAIMS which is a direction not used in CAPRI

// up to this point. The audit log will login to CLAIMS, but that is not done

// on a test account.

//------------------------------------------------------------------------------

// TEST CLAIMS DATA:

// FILE: 396.97, FIELD .01

// @DEV/FEX TEST SYSTEM

// ZZ 553DT

// KANSAS CITY(DSS)

// BAY PINES-GC

// BAY PINES-PRRTP

// ZZ WEST PLAINS CBOC

// CHEYENNE-VBA

// BAY PINES-CIOFO

// BAY PINES NURSING HOME

// CHEY133.DNS

// ORLANDO VAMC (EFF. 10/1/06)

// CHEY133\_LINUX.DNS

// ZZ BEST

// ALASKA VAHSRO

// IRON MOUNTAIN VAMC

// CHYSHR

// DAYTSHR

// CHEYL147

// CHEYL135

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestGuestLoginClick(Sender: TObject);

var

tempServer : String;

tempPort : Integer;

tempTestMode : Boolean;

const

TESTCLAIMSSERVER = 'VISTA.DNS .MED.DNS ';

TESTPORT = '9401';

begin

Application.ProcessMessages;

try

Screen.Cursor := crHourGlass;

StatusBar1.SimpleText := 'Connecting to TEST CLAIMS server';

tempServer := frmMain.RPCBroker1.Server;

tempPort := frmMain.RPCBroker1.ListenerPort;

tempTestMode := frmMain.RPCBroker1.ANUTestMode;

frmMain.RPCBroker1.Connected := False;

frmMain.RPCBroker1.Server := TESTCLAIMSSERVER;

frmMain.RPCBroker1.ListenerPort := StrToInt(TESTPORT);

frmMain.RPCBroker1.ANUTestMode := TRUE;

if not frmMain.ConnectToServer('DVBA CAPRI GUI') then

begin

ShowMessageCAPRI('Cannot connect to ' + TESTCLAIMSSERVER);

exit;

end;

StatusBar1.SimpleText := 'Connected to TEST CLAIMS server';

// Populate list with Fileman control data

FMExamList.Clear;

FMDivisionComments.Clear;

// Populate list with RPC data

StatusBar1.SimpleText := 'Get Fileman Control Data.';

FMExamList.FMLister := FMReRouteLister;

FMReRouteLister.GetList(FMExamList.Items);

StatusBar1.SimpleText := 'Get RPC Data.';

frmMain.RPCBroker1.RemoteProcedure := 'DVBA CHECK PATCH';

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := 'DVBA\*2.7\*190';

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

if frmMain.RPCBroker1.Results.Count > 0 then

if piece(frmMain.RPCBroker1.Results[0], '^', 1) = '1' then

FMDivisionComments.Lines.Add('Patch 190 installed')

else

FMDivisionComments.Lines.Add('Patch 190 in not installed')

except on E: Exception do

FMDivisionComments.Lines.Add('Exception on GetData: ' + E.Message);

end;

finally

StatusBar1.SimpleText := 'Disconnect from TEST CLAIMS. Reconnect to local server.';

frmMain.RPCBroker1.Connected := False;

frmMain.RPCBroker1.Server := TempServer;

frmMain.RPCBroker1.ANUTestMode := tempTestMode;

frmMain.RPCBroker1.ListenerPort := TempPort;

frmMain.ConnectToServer('DVBA CAPRI GUI');

StatusBar1.SimpleText := 'Reconnected to local server.';

Screen.Cursor := crDefault;

end;

end;

Table 48: Functions

| Function Name | FMVAMCExit | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Selected VAMC. Populate Routing locations drop-down. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FMVAMCExit | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Select VAMC

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.FMVAMCExit(Sender: TObject);

var

buffer : String;

begin

Application.ProcessMessages;

if FMVAMC.ItemIndex <> -1 then

VAMC\_IEN := VAMC\_IEN\_list[FMVAMC.ItemIndex]

else

exit;

SelectedStation := VAMC\_Station\_List[FMVAMC.ItemIndex];

// If we can't re-route to the selected VAMC, set the VAMC to blank and

// give the user a message.

if CheckIfVAMCCanHandleReroutes = FALSE then

begin

buffer := Copy(FMVAMC.Text,1,28);

FMVAMC.ItemIndex := -1;

ShowMessageCAPRI(trim(buffer) + ' cannot accept re-routed requests at this time.');

FMVAMC.SetFocus;

end

else

GetRoutingLocations;

end;

Table 48: Functions

| Function Name | cbxLocationChange | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Selected Routing Location. Populated exam list and site comments. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | cbxLocationChange | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Select Routing Location

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.cbxLocationChange(Sender: TObject);

begin

Application.ProcessMessages;

if cbxLocation.ItemIndex <> -1 then

begin

RoutingLocation\_IEN := RoutingLocation\_IEN\_List[cbxLocation.ItemIndex];

StatusBar1.SimpleText := 'Retrieving Exam Info';

GetExamLists;

GetSiteComments;

StatusBar1.SimpleText := '';

end;

end;

Table 48: Functions

| Function Name | FormClose | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Close the form. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormClose | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Form Close

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.FormClose(Sender: TObject;

var Action: TCloseAction);

begin

// Destroy string lists

FreeAndNil(VAMC\_IEN\_List);

FreeAndNil(RoutingLocation\_IEN\_List);

FreeAndNil(VAMC\_Station\_List);

FreeAndNil(VAMC\_Name\_List);

FreeAndNil(VAMC\_State\_List);

FreeAndNil(ReRouteReasonIENList);

end;

Table 48: Functions

| Function Name | FormCreate | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Setup the form. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormCreate | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Form Create

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.FormCreate(Sender: TObject);

begin

// Display testing buttons if Delphi IDE is running. This is only for

// developer testing: btnTestRoutingLocations, btnTestGuestLogin,

// btnTestRemoteRPC.

IsDelphiRunning;

// Create string lists

VAMC\_IEN\_List := TStringList.Create;

RoutingLocation\_IEN\_List := TStringList.Create;

VAMC\_Station\_List := TStringList.Create;

VAMC\_Name\_List := TStringList.Create;

VAMC\_State\_List := TStringList.Create;

ReRouteReasonIENList := TStringList.Create;

// Populate initial fields

GetReRouteReasons; // Populate Re-Routing Reasons dropdown

GetVAMCLocations; // Populate VAMC Locations dropdown

end;

Table 48: Functions

| Function Name | btnTestRemoteRPCClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnTestRemoteRPCClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// TEST ROUTINE - XWB DIRECT RPC

// -----------------------------------------------------------------------------

// This routine calls an RPC that uses HL7 to communicate between VistA

// instances. It was exactly what we needed, but our test accounts do not have

// this HL7 interface installed. In addition, TEST CLAIMS does not have a

// unique station number that is published that also causes issues. Leaving

// this routine here in case we come back to it.

//

// Update: HL7 was added to our test environments between DEVFEX and TEST

// CLAIMS. Other dev environments may be added. Our dev environment has

// unique station numbers between the systems it knows about (only dev, not

// production) so HL7 can communicate. HL7 is currently deployed in all

// production environments so no worries about using the RPC in production.

// -----------------------------------------------------------------------------

// XWB DIRECT RPC

// Param 0: Station Number

// Param 1: Name of RPC

// Param 2: RPC VERSION PARAMETER (Optional)

// Param 3: Remote RPC Parameters

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestRemoteRPCClick(Sender: TObject);

const

RPCNAME = 'DVBA CHECK PATCH';

PATCH = 'DVBA\*2.7\*193';

var

ReturnValue : String;

begin

FMExamList.Items.Clear;

frmMain.RPCBroker1.RemoteProcedure := 'XWB DIRECT RPC';

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := edtStationNumber.Text;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := '';

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := PATCH;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

if frmMain.RPCBroker1.Results.Count > 0 then

QuickCopy(frmMain.RPCBroker1.Results, FMExamList.Items);

except

ShowMessage('Exception');

end;

end;

Table 48: Functions

| Function Name | btnTestRoutingLocationsClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnTestRoutingLocationsClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// TEST ROUTINE - TEST ROUTING LOCATIONS

// -----------------------------------------------------------------------------

// This routine calls DVBA CAPRI GET DIVISION using the station number input in

// the TEST STATION / DIVISION input field. Only used during development on

// the local station number.

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestRoutingLocationsClick(Sender: TObject);

const

RPCNAME = 'DVBA CAPRI GET DIVISION';

var

i : Integer;

begin

StatusBar1.SimpleText := 'Retrieving Routing Locations.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

cbxLocation.Items.Clear;

RoutingLocation\_IEN\_List.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := edtStationNumber.Text;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

begin

For i := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

begin

cbxLocation.Items.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 1));

RoutingLocation\_IEN\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 2));

end;

end;

except

On EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | btnTestExamListClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnTestExamListClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// TEST ROUTINE - TEST EXAM LIST

// -----------------------------------------------------------------------------

// Calls DVBA CAPRI GET DIV EXAM from the local VistA instance. The value

// input in the TEST STATION/DIVISION input should be a local division IEN.

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestExamListClick(Sender: TObject);

CONST

RPCNAME = 'DVBA CAPRI GET DIV EXAM';

DIVISION = '3'; // ALBANY ON DEVFEX

begin

StatusBar1.SimpleText := 'Retrieving Exam Lists.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMExamList.Items.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := DIVISION;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

QuickCopy(frmMain.RPCBroker1.Results, FMExamList.Items);

except

On EBrokerError Do

begin

Screen.Cursor := crHourGlass;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Division ' + DIVISION + '!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | btnTestSiteCommentsClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | btnTestSiteCommentsClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// TEST ROUTINE - TEST SITE COMMENTS

// -----------------------------------------------------------------------------

// Calls DVBA CAPRI GET DIV COMMENT from the local VistA instance. The value

// input in the TEST STATION/DIVISION input should be a local division IEN.

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestSiteCommentsClick(Sender: TObject);

CONST

RPCNAME = 'DVBA CAPRI GET DIV COMMENT';

DIVISION = '3'; // ALBANY ON DEVFEX

var

i : Integer;

begin

StatusBar1.SimpleText := 'Retrieving Site Comments.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMDivisionComments.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].Value := DIVISION;

frmMain.RPCBroker1.Param[0].PType := literal;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

QuickCopy(frmMain.RPCBroker1.Results, FMExamList.Items);

except

On EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Division ' + DIVISION + '!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | cbxSortByStateClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Sort by state checkbox clicked. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | cbxSortByStateClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

// -----------------------------------------------------------------------------

// TEST ROUTINE - TEST SITE COMMENTS

// -----------------------------------------------------------------------------

// Calls DVBA CAPRI GET DIV COMMENT from the local VistA instance. The value

// input in the TEST STATION/DIVISION input should be a local division IEN.

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.btnTestSiteCommentsClick(Sender: TObject);

CONST

RPCNAME = 'DVBA CAPRI GET DIV COMMENT';

DIVISION = '3'; // ALBANY ON DEVFEX

var

i : Integer;

begin

StatusBar1.SimpleText := 'Retrieving Site Comments.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMDivisionComments.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].Value := DIVISION;

frmMain.RPCBroker1.Param[0].PType := literal;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

QuickCopy(frmMain.RPCBroker1.Results, FMExamList.Items);

except

On EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Division ' + DIVISION + '!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | CreateReRouteRequest | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Setup parameters and call RPC to create a re-routed request. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | CreateReRouteRequest | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Create Re-Route Request

//------------------------------------------------------------------------------

// RPC Input:

// Request IEN

// Patient IEN

// VAMC IEN (Station Number)

// Routing Location IEN

// Re-Route Reason IEN

// Re-Route Description

//

// RPC Output:

// 1^Transmitted as message # 909761 from this site to DEVVCC.DNS .MED.DNS

// 0^Message transmission error! Request WILL NOT be rerouted

// -----------------------------------------------------------------------------

// FROM THE AMIE TRANSFER DOCUMENTATION:

// If the mail message is sent out successfully, the user will see:

// "Transmitted as message # {number} from this site to {site name}.DNS ."

//

// If the message fails (and therefore the transfer fails) the user will be so informed:

// "Message transmission error! Request WILL NOT be sent !"

// -----------------------------------------------------------------------------

function TfrmReRouteRequest.CreateReRouteRequest : Boolean;

CONST

RPCNAME = 'DVBA CAPRI SEND REROUTE';

var

MailmanRPCStatus : Integer;

MailmanMessage : String;

begin

Result := FALSE; // default to re-route failed

try

Screen.Cursor := crHourGlass;

// Setup RPC

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].Value := frmViewExam.edExamReferenceNumber.Text; // Request IEN

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[1].Value := PatientIEN; // Patient IEN

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[2].Value := VAMC\_IEN;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[3].Value := RoutingLocation\_IEN;

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[4].Value := ReRouteReasonIENList[FMReRouteReason.ItemIndex];

frmMain.RPCBroker1.Param[4].PType := literal;

frmMain.RPCBroker1.Param[5].Value := FMReRouteDescription.Text;

frmMain.RPCBroker1.Param[5].Ptype := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On e: EBrokerError Do

begin

ShowMessageCAPRI(RPCNAME + ' could not be accessed! '+#10#13 + E.Message + '.');

end;

End;

finally

Screen.Cursor := crDefault;

end;

// Check RPC results

If frmMain.RPCBroker1.Results.Count > 0 Then

begin

MailmanRPCStatus := StrToIntDef(Piece(frmMain.RPCBroker1.Results[0], '^', 1),0);

MailmanMessage := Piece(frmMain.RPCBroker1.Results[0], '^', 2);

StatusBar1.SimpleText := MailmanMessage;

if MailmanRPCStatus > 0 then // Success, request was re-routed

begin

// FYI: An email is sent to the original requester upon successful re-routing (all M code)

ShowMessageCAPRI(MailmanMessage);

Result := TRUE;

Close;

end

end;

// if Re-Route message failed

if not Result then

ShowMessageCAPRI(RPCNAME + ' returned a failure.'+#10#13+MailmanMessage);

end;

Table 48: Functions

| Function Name | CheckUserInputValues | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Check input values to make certain all values are completed before re-routing a request. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | CheckUserInputValues | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Check User Input Values

//------------------------------------------------------------------------------

// Check all user input values. If values are input correctly, return TRUE.

// If values are missing, routine short circuits back to input form.

//------------------------------------------------------------------------------

function TfrmReRouteRequest.CheckUserInputValues: Boolean;

begin

if FMReRouteReason.Text = '' then

begin

ShowMessageCAPRI('Please enter a Re-Routing Reason.');

Result := FALSE;

exit; // make user complete form before sending

end;

if FMReRouteReason.Text = 'OTHER' then

begin

if Length(FMReRouteDescription.Lines.Text) = 0 then

begin

ShowMessageCAPRI('Please enter a Re-Routing Description.');

Result := FALSE;

exit; // make user complete form before sending

end;

end;

if FMVAMC.ItemIndex = -1 then

begin

ShowMessageCAPRI('Please enter a VAMC.');

Result := FALSE;

exit; // make user complete form before sending

end;

if cbxLocation.ItemIndex = -1 then

begin

ShowMessageCAPRI('Please enter a Routing Location.');

Result := FALSE;

exit; // make user complete form before sending

end;

Result := TRUE;

end;

Table 48: Functions

| Function Name | GetRoutingLocations | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Retrieve the Routing Locations from the remote VistA instance selected in the VAMC list. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetRoutingLocations | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Get Routing Locations

//------------------------------------------------------------------------------

// Call RPC on remote VistA site specified by Station Number passed to RPC.

// Input:

// RPC: XWB DIRECT RPC

// RPC: Station Number of remote VistA instance (VAMC)

// Local RPC: DVBA CAPRI GET DIVISION

// Local RPC Version: Not specified here, sending blank

// Local RPC Parameter: Station Number

// Output: List of Routing Locations (Divisions) for the input Station Number

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.GetRoutingLocations;

CONST

RPC\_DIRECT = 'XWB DIRECT RPC';

RPCNAME = 'DVBA CAPRI GET DIVISION';

var

i : Integer;

begin

StatusBar1.SimpleText := 'Retrieving Routing Locations.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMExamList.Items.Clear;

// determine if we are re-routing locally versus to a remote VistA site

if frmViewExam.CurrentStationNumber = SelectedStation then

begin // Call RPC locally

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := SelectedStation;

end

else

begin // Call RPC via XWB DIRECT RPC

frmMain.RPCBroker1.RemoteProcedure := RPC\_DIRECT;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := SelectedStation;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := ''; // RPC version parameter (Optional)

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := SelectedStation;

end;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

cbxLocation.Items.Clear;

RoutingLocation\_IEN\_List.Clear;

If frmMain.RPCBroker1.Results.Count > 0 Then

begin

For i := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

begin

cbxLocation.Items.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 1));

RoutingLocation\_IEN\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 2));

end;

end;

except

On E: EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Station Number ' + SelectedStation + '!' +#13#10 + E.Message);

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | GetVAMCLocations | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Retrieve the VAMC list. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetVAMCLocations | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Get VAMC Locations

//------------------------------------------------------------------------------

// VAMCs are located in the Institution File. Filtering and sorting must be

// done to get an accurate selection list.

// -----------------------------------------------------------------------------

// RPC Input: Sorting Flag (NAME or STATE)

// RPC Output: State ^ VAMC sites ^ VAMC IEN ^ VAMC Station Number

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.GetVAMCLocations;

var

i : Integer;

CONST

RPCNAME = 'DVBA CAPRI REROUTE VAMC';

STRINGFORMAT = '%-28s%-15s%-6s';

begin

// update status

StatusBar1.SimpleText := 'Retrieving Routing Locations.';

// setup and call RPC

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

if cbxSortByState.Checked then

frmMain.RPCBroker1.Param[0].Value := 'STATE'

else

frmMain.RPCBroker1.Param[0].Value := 'NAME';

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On E: EBrokerError Do

ShowMessageCAPRI(RPCNAME + ' could not be accessed!'+#13#10+E.Message);

End;

// clear all lists

FMVAMC.Items.Clear;

VAMC\_Name\_List.Clear;

VAMC\_State\_List.Clear;

VAMC\_IEN\_List.Clear;

VAMC\_Station\_List.Clear;

// populate all lists

If frmMain.RPCBroker1.Results.Count > 0 Then

begin

For i := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

begin

VAMC\_Name\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 2));

VAMC\_State\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 1));

VAMC\_IEN\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 3));

VAMC\_Station\_List.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 4));

end;

end;

// Add data to listbox and format it into columns

for i := 0 to VAMC\_Name\_List.Count - 1 do

FMVAMC.Items.Add(Format(STRINGFORMAT,[VAMC\_Name\_List[i],VAMC\_State\_List[i],VAMC\_Station\_List[i]]));

// clear status bar

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | GetExamLists | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Retrieve the exam lists from the remote VistA instance selected in the VAMC list. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetExamLists | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Get Exam Lists

//------------------------------------------------------------------------------

// Input:

// RPC: XWB DIRECT RPC

// RPC: Station Number of remote VistA instance (VAMC)

// Local RPC: DVBA CAPRI GET DIV EXAM

// Local RPC Version: Not specified here, sending blank

// Local RPC Parameter: Routing Location (Division) IEN

// Output: List of Exams Available at selected site

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.GetExamLists;

CONST

RPC\_DIRECT = 'XWB DIRECT RPC';

RPCNAME = 'DVBA CAPRI GET DIV EXAM';

var

i : Integer;

begin

StatusBar1.SimpleText := 'Retrieving Exam Lists.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMExamList.Items.Clear;

if frmViewExam.CurrentStationNumber = SelectedStation then

begin

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := RoutingLocation\_IEN; // routing location (division) selected

end

else

begin

frmMain.RPCBroker1.RemoteProcedure := RPC\_DIRECT;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := SelectedStation;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := ''; // RPC version parameter (Optional)

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := RoutingLocation\_IEN; // routing location (division) selected

end;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

QuickCopy(frmMain.RPCBroker1.Results, FMExamList.Items);

except

On EBrokerError Do

begin

Screen.Cursor := crHourGlass;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Station Number ' + SelectedStation + '!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | GetSiteComments | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Retrieve the site comments from the remote VistA instance selected in the VAMC list. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetSiteComments | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Get Site Comments

//------------------------------------------------------------------------------

// Routing Locations are located in the Institution File. Filtering and sorting

// must be done to get an accurate selection list.

// -----------------------------------------------------------------------------

// Input:

// RPC: XWB DIRECT RPC

// RPC: Station Number of remote VistA instance (VAMC)

// Local RPC: DVBA CAPRI GET DIV COMMENT

// Local RPC Version: Not specified here, sending blank

// Local RPC Parameter: Routing Location (Division) IEN

// Output: List of Re-Route sites

// -----------------------------------------------------------------------------

procedure TfrmReRouteRequest.GetSiteComments;

CONST

RPC\_DIRECT = 'XWB DIRECT RPC';

RPCNAME = 'DVBA CAPRI GET DIV COMMENT';

begin

StatusBar1.SimpleText := 'Retrieving Site Comments.';

Screen.Cursor := crHourGlass;

Application.ProcessMessages;

FMDivisionComments.Lines.Clear;

if frmViewExam.CurrentStationNumber = SelectedStation then

begin

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := RoutingLocation\_IEN; // routing location (division) selected

end

else

begin

frmMain.RPCBroker1.RemoteProcedure := RPC\_DIRECT;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := SelectedStation;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := ''; // RPC version parameter (Optional)

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := RoutingLocation\_IEN; // routing location (division) selected

end;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

QuickCopy(frmMain.RPCBroker1.Results, FMDivisionComments.Lines);

except

On EBrokerError Do

begin

Screen.Cursor := crDefault;

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Station Number ' + SelectedStation + '!');

end;

end;

Screen.Cursor := crDefault;

StatusBar1.SimpleText := '';

end;

Table 48: Functions

| Function Name | GetReRouteReasons | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Retrieve the Re-Route Reasons and populate the drop-down list. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetReRouteReasons | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Get Re-Route Reasons

//------------------------------------------------------------------------------

// RPC: DVBA CAPRI REROUTEREASONS

// Input: None

// Output: Re-Route Code^IEN

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.GetReRouteReasons;

const

RPCNAME = 'DVBA CAPRI GET REROUTE CODE';

var

i : Integer;

begin

frmMain.RPCBroker1.Results.Clear;

frmMain.RPCBroker1.RemoteProcedure := RPCNAME;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

On EBrokerError Do

ShowMessageCAPRI('RPC: '+RPCNAME + ' could not be accessed!');

end;

FMReRouteReason.Items.Clear;

ReRouteReasonIENList := TStringList.Create;

if frmMain.RPCBroker1.Results.Count > 0 then

begin

for i := 0 To frmMain.RPCBroker1.Results.Count - 1 do

begin

FMReRouteReason.Items.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 2));

ReRouteReasonIENList.Add(Piece(frmMain.RPCBroker1.Results[i], '^', 1));

end;

end;

end;

Table 48: Functions

| Function Name | IsDelphiRunning | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Debug Routine. Not used. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | IsDelphiRunning | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Is Delphi Running?

//------------------------------------------------------------------------------

// If Delphi is running, we are in the IDE and operating in a testing mode.

// Display testing buttons. If Delphi is not running, do not display test

// buttons.

//------------------------------------------------------------------------------

procedure TfrmReRouteRequest.IsDelphiRunning;

begin

if DebugHook <> 0 then

begin

btnTestGuestLogin.Visible := TRUE;

btnTestRemoteRPC.Visible := TRUE;

btnTestRoutingLocations.Visible := TRUE;

btnTestExamList.Visible := TRUE;

btnTestSiteComments.Visible := TRUE;

edtStationNumber.Visible := TRUE;

end

else

begin

btnTestGuestLogin.Visible := FALSE;

btnTestRemoteRPC.Visible := FALSE;

btnTestRoutingLocations.Visible := FALSE;

btnTestExamList.Visible := FALSE;

btnTestSiteComments.Visible := FALSE;

edtStationNumber.Visible := FALSE;

end;

end;

Table 48: Functions

| Function Name | CheckIfVAMCCanHandleReroutes | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Check if the VAMC selected has Patch 193 installed before allowing a re-route. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | CheckIfVAMCCanHandleReroutes | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

NEW

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

//------------------------------------------------------------------------------

// Check if VAMC Can Handle Re-Routes

//------------------------------------------------------------------------------

// If the destination site doesn't have Patch 193 installed, then it will not

// know how to receive a re-routed request. Don't allow the user to re-route

// to a site where Patch 193 isn't installed.

// Input: Station Number at distant site

// Output:

// TRUE: ok to re-route

// FALSE: cannot re-route to this site

// -----------------------------------------------------------------------------

function TfrmReRouteRequest.CheckIfVAMCCanHandleReroutes: Boolean;

const

RPC\_DIRECT = 'XWB DIRECT RPC';

RPCNAME = 'DVBA CHECK PATCH';

PATCHNUM = 'DVBA\*2.7\*193';

begin

Result := FALSE;

// if ItemIndex is not selected (or forced to -1), don't test. Just fail

if FMVAMC.ItemIndex = -1 then

exit;

// if we are re-routing within the same VAMC, do not check the patch.

// if we got this far, the patch is installed.

if SelectedStation = frmViewExam.CurrentStationNumber then

begin

Result := TRUE;

Exit;

end;

try

Screen.Cursor := crHourGlass;

// check patch number at selected site. If Patch 193 is not installed, error

frmMain.RPCBroker1.RemoteProcedure := RPC\_DIRECT;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBroker1.Param[0].Value := SelectedStation;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[1].Value := RPCNAME;

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[2].Value := ''; // RPC version parameter (Optional)

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBroker1.Param[3].Value := PATCHNUM;

try

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

if frmMain.RPCBroker1.Results.Count > 0 then

if piece(frmMain.RPCBroker1.Results[0], '^', 1) = '1' then

Result := TRUE;

except

begin

ShowMessageCAPRI(RPCNAME + ' could not be accessed at Station Number ' + SelectedStation + '!');

Result := FALSE;

end

end;

finally

Screen.Cursor := crDefault;

end;

end;

##### GUI – DemTranMain.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| DemTranMain.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events – N/A for Requirement 2.6.1

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| UpdateVistaSFTP | Function | Modified to handle Request Status database field change. |

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause

CapriSupport

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | UpdateVistaSFTP | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | UpdateVistaSFTP | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

function TfrmCntrctdExmMain.UpdateVistaSFTP(slResults, slOurExams: TStringList; ReqIen,

ExamIen, IntResDt, ResultStub: String; ReqAllDone: Boolean): Boolean;

var

ResultsMemo: TFMMemo;

ResultsFiler, ReqFiler: TFMFiler;

ExamRef, ExamTypPtr, X: String;

i: Integer;

FirstOne: Boolean;

begin

Result := True;

ResultsMemo := TFMMemo.Create(nil); // DONE: LMS all these creates don't pair with finally. Possible memory leak.

ResultsMemo.ParentWindow := frmCntrctdExmMain.Handle;

ResultsFiler := TFMFiler.Create(nil);

ReqFiler := TFMFiler.Create(nil);

try

// DONE: LMS - no - non existing member of class.. what functionality is lost?

ResultsFiler.RPCBroker := CCOWRPCBrkrMain;

ReqFiler.RPCBroker := CCOWRPCBrkrMain;

ResultsFiler.FilerFlags := [ffLock];

ResultsFiler.FilerOptions := [foProcessProblemList];

ReqFiler.FilerFlags := [ffLock];

ReqFiler.FilerOptions := [foProcessProblemList];

ResultsMemo.FMFiler := ResultsFiler;

ResultsMemo.FMFile := '396.4';

ResultsMemo.FMField := '70';

ResultsMemo.IENS := ExamIen;

ResultsMemo.FMMemoStyle := msByLines; // DONE: LMS Contractor supplied results were getting wrapped.

ResultsMemo.Lines := slResults;

ResultsMemo.FMModified := True;

ResultsFiler.AddChgdControl(ResultsMemo);

if ResultsFiler.Update then

StatusMsg('Results filed under exam #' + ExamIen)

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

i := 0;

while (i < slOurExams.Count) do

begin

X := Piece(slOurExams[i], '-', 3);

ResultsMemo.IENS := X;

ExamRef := Piece(slOurExams[i], '-', 4);

ExamTypPtr := Piece(slOurExams[i], '-', 5);

FirstOne := (CompareText(X, ExamIen) = 0);

if (not FirstOne) then

begin

ResultsMemo.Lines.Clear;

ResultsMemo.Lines.Add(ResultStub);

ResultsFiler.AddChgdControl(ResultsMemo);

if ResultsFiler.Update then

StatusMsg('Results stub message for exam ' + X + ' filed')

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

end;

ResultsFiler.AddFDA('396.4', X + ',', '.01', ExamRef);

ResultsFiler.AddFDA('396.4', X + ',', '.03', ExamTypPtr);

ResultsFiler.AddFDA('396.4', X + ',', '.04', 'C');

// ResultsFiler.AddFDA('396.4', X + ',', '90', IntResDt); // DONE: LMS CodeCR533 Commenting this line out because updating this field is being seen as setting that the exam is "Transcribed"Date" and as a result not allowed to "auto-release". See code referring to field 90 in tiusign.pas and uncosignedutility.pas units... also see referred code section below.

// further.. nowhere in the production demtran version 1.3.2.6 source code did I find a reference to field 90

(\*

if FMGetsExamInfo.GetField('90').FMDBExternal <> '' then

Transcribed := True;

\*)

// below line updating 102 is already there. Will point references to 90 to use 102 instead. One reference is in managereportsCCR.pas .. see below

(\*

FMfiler1.AddFDA('396.4', whichIEN, '.90', '');

\*)

// 102I is being labeled Date received from contractor. How's that different from 102

ResultsFiler.AddFDA('396.4', X + ',', '102', IntResDt);

ResultsFiler.AddFDA('396.4', X + ',', '73', '@'); // 396.4,73 XML DAS CONFIRMATION 4;1 FREE TEXT. "Description: This is the DAS status confirmation returned by VLER DAS.". Storing DAS result object id here for VLER only. Clearing for SFTP

if ResultsFiler.Update then

StatusMsg('Fields updated for exam #' + X)

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

Inc(i);

end;

if (ReqAllDone) then

begin

ReqFiler.AddFDA('396.3', ReqIen, '17', 'P');

if ReqFiler.Update then begin

StatusMsg('Request #' + ReqIen + 'updated.');

end else begin

ReqFiler.DisplayErrors;

Result := False;

end;

end;

// StatusMsg('');

finally

ResultsMemo.Free;

ResultsFiler.Free;

ReqFiler.Free;

end;

// Result := True; // DONE: LMS only if successful! Added lines above in method to appropriately return value.

end;

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

function TfrmCntrctdExmMain.UpdateVistaSFTP(slResults, slOurExams: TStringList; ReqIen,

ExamIen, IntResDt, ResultStub: String; ReqAllDone: Boolean): Boolean;

var

ResultsMemo: TFMMemo;

ResultsFiler, ReqFiler: TFMFiler;

ExamRef, ExamTypPtr, X: String;

i: Integer;

FirstOne: Boolean;

begin

Result := True;

ResultsMemo := TFMMemo.Create(nil); // DONE: LMS all these creates don't pair with finally. Possible memory leak.

ResultsMemo.ParentWindow := frmCntrctdExmMain.Handle;

ResultsFiler := TFMFiler.Create(nil);

ReqFiler := TFMFiler.Create(nil);

try

// DONE: LMS - no - non existing member of class.. what functionality is lost?

ResultsFiler.RPCBroker := CCOWRPCBrkrMain;

ReqFiler.RPCBroker := CCOWRPCBrkrMain;

ResultsFiler.FilerFlags := [ffLock];

ResultsFiler.FilerOptions := [foProcessProblemList];

ReqFiler.FilerFlags := [ffLock];

ReqFiler.FilerOptions := [foProcessProblemList];

ResultsMemo.FMFiler := ResultsFiler;

ResultsMemo.FMFile := '396.4';

ResultsMemo.FMField := '70';

ResultsMemo.IENS := ExamIen;

ResultsMemo.FMMemoStyle := msByLines; // DONE: LMS Contractor supplied results were getting wrapped.

ResultsMemo.Lines := slResults;

ResultsMemo.FMModified := True;

ResultsFiler.AddChgdControl(ResultsMemo);

if ResultsFiler.Update then

StatusMsg('Results filed under exam #' + ExamIen)

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

i := 0;

while (i < slOurExams.Count) do

begin

X := Piece(slOurExams[i], '-', 3);

ResultsMemo.IENS := X;

ExamRef := Piece(slOurExams[i], '-', 4);

ExamTypPtr := Piece(slOurExams[i], '-', 5);

FirstOne := (CompareText(X, ExamIen) = 0);

if (not FirstOne) then

begin

ResultsMemo.Lines.Clear;

ResultsMemo.Lines.Add(ResultStub);

ResultsFiler.AddChgdControl(ResultsMemo);

if ResultsFiler.Update then

StatusMsg('Results stub message for exam ' + X + ' filed')

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

end;

ResultsFiler.AddFDA('396.4', X + ',', '.01', ExamRef);

ResultsFiler.AddFDA('396.4', X + ',', '.03', ExamTypPtr);

ResultsFiler.AddFDA('396.4', X + ',', '.04', 'C');

// ResultsFiler.AddFDA('396.4', X + ',', '90', IntResDt); // DONE: LMS CodeCR533 Commenting this line out because updating this field is being seen as setting that the exam is "Transcribed"Date" and as a result not allowed to "auto-release". See code referring to field 90 in tiusign.pas and uncosignedutility.pas units... also see referred code section below.

// further.. nowhere in the production demtran version 1.3.2.6 source code did I find a reference to field 90

(\*

if FMGetsExamInfo.GetField('90').FMDBExternal <> '' then

Transcribed := True;

\*)

// below line updating 102 is already there. Will point references to 90 to use 102 instead. One reference is in managereportsCCR.pas .. see below

(\*

FMfiler1.AddFDA('396.4', whichIEN, '.90', '');

\*)

// 102I is being labeled Date received from contractor. How's that different from 102

ResultsFiler.AddFDA('396.4', X + ',', '102', IntResDt);

ResultsFiler.AddFDA('396.4', X + ',', '73', '@'); // 396.4,73 XML DAS CONFIRMATION 4;1 FREE TEXT. "Description: This is the DAS status confirmation returned by VLER DAS.". Storing DAS result object id here for VLER only. Clearing for SFTP

if ResultsFiler.Update then

StatusMsg('Fields updated for exam #' + X)

else begin

ResultsFiler.DisplayErrors;

Result := False;

end;

Inc(i);

end;

if (ReqAllDone) then

begin

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**ReqFiler.AddFDA('396.3', ReqIen, '17', '2') // Patch 193**

**else // Patch 193**

ReqFiler.AddFDA('396.3', ReqIen, '17', 'P'); // Patch 193

if ReqFiler.Update then begin

StatusMsg('Request #' + ReqIen + 'updated.');

end else begin

ReqFiler.DisplayErrors;

Result := False;

end;

end;

// StatusMsg('');

finally

ResultsMemo.Free;

ResultsFiler.Free;

ReqFiler.Free;

end;

// Result := True; // DONE: LMS only if successful! Added lines above in method to appropriately return value.

end;

##### GUI – frmContractedExamCheckIn.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| frmContractedExamCheckIn.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events – N/A for Requirement 2.6.1

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| CheckInExams | Function | Modified to handle Request Status database field change. |

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause

CapriSupport

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | CheckInExams | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | CheckInExams | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

function TExntdCntrctdBaseFormCheckIn.CheckInExams(exmObjLst: TStringList;

allExmsCmpltdCncld: Boolean; OverWriteRslts: Boolean; VRslts: TStrings): Boolean;

const

ADDTNLEXMRSLTS = 'Please refer to the exam report contained in: ';

var

i: Integer;

err: Boolean;

aExmRslts: TStrings;

exmIEN, exmNme, chkInDTM: String;

begin

err := False;

i := -1;

chkInDtm := DelphiDateToFMInternal(StrToDate(edDtPkrCheckIn.Text));

try

repeat

inc(i);

exmIEN := exmObjLst[i];

fmMemoExmRslts.Lines.Clear;

{ Set Memo to Exam's Current results (Could be empty) if NOT Overwriting }

if Not(OverWriteRslts) then

fmMemoExmRslts.Lines.Assign(TExamInfo(exmObjLst.Objects[i]).GetExamResults);

{ Save off Exam Name in case mulitple exams selected for check-in and

Add Vendor Exam Results Info or pointer to exam results in to TFMMemo.

More than 1 Exam selected will cause remaining exam results to point

back to the initial exam that has the complete vendor exam results. }

if (i = 0) then

begin

exmNme := \_fmTmpLstrExams.GetRecord(exmIEN).GetField('.03').FMDBExternal;

AddVendorRsltsToExmRslts(VRslts); { Only add results to initial exam }

end

else

begin

aExmRslts := TStringList.Create;

aExmRslts.Add(ADDTNLEXMRSLTS);

aExmRslts.Add(exmNme);

AddVendorRsltsToExmRslts(aExmRslts);

aExmRslts.Clear;

aExmRslts.Free;

end; { Add Pointer Exam Results to Memo for subsequent selected exams }

try

err := Not \_2507Info.CheckInExam(exmIEN, chkInDTM, fmMemoExmRslts);

except

on E: Exception do

begin

err := True; { So any previously modified exams are restored }

{ Handle Exception where the length of one or more lines in the Vendors

completed exam text file to be uploaded exceeds 999 characters in length. }

raise EVndrExmLngthError.Create('The length of one or more lines in the Vendors ASCII text file ' +

'is greater than 999 characters.' + #13#10 + 'Contact the Vendor to ' +

'have the file(s) corrected before trying to complete the check-in for this request again.');

end; { End Exception Handling }

end; { End Inner Try/Except Block }

until ((i = (exmObjLst.Count - 1)) or err);

{ No Errors and All Exams Completed/Cancelled - Set Request Status to 'P' }

if (Not err) and allExmsCmpltdCncld then

begin

\_2507Info.UpdtRqstStatus(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, 'P');

end;

finally

if err then { Error - restore modified exams to original state }

for i := 0 to exmObjLst.Count - 1 do

begin

exmIEN := exmObjLst[i];

{ Retrieve ORIGINAL exam results to restore }

fmMemoExmRslts.Lines.Assign(TExamInfo(exmObjLst.Objects[i]).GetExamResults);

\_2507Info.CheckInExam(exmIEN, '@', fmMemoExmRslts);

end; { End Modified Exam List Loop }

end; { End Outer Try/Finally Block }

Result := Not err;

end;

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

function TExntdCntrctdBaseFormCheckIn.CheckInExams(exmObjLst: TStringList;

allExmsCmpltdCncld: Boolean; OverWriteRslts: Boolean; VRslts: TStrings): Boolean;

const

ADDTNLEXMRSLTS = 'Please refer to the exam report contained in: ';

var

i: Integer;

err: Boolean;

aExmRslts: TStrings;

exmIEN, exmNme, chkInDTM: String;

begin

err := False;

i := -1;

chkInDtm := DelphiDateToFMInternal(StrToDate(edDtPkrCheckIn.Text));

try

repeat

inc(i);

exmIEN := exmObjLst[i];

fmMemoExmRslts.Lines.Clear;

{ Set Memo to Exam's Current results (Could be empty) if NOT Overwriting }

if Not(OverWriteRslts) then

fmMemoExmRslts.Lines.Assign(TExamInfo(exmObjLst.Objects[i]).GetExamResults);

{ Save off Exam Name in case mulitple exams selected for check-in and

Add Vendor Exam Results Info or pointer to exam results in to TFMMemo.

More than 1 Exam selected will cause remaining exam results to point

back to the initial exam that has the complete vendor exam results. }

if (i = 0) then

begin

exmNme := \_fmTmpLstrExams.GetRecord(exmIEN).GetField('.03').FMDBExternal;

AddVendorRsltsToExmRslts(VRslts); { Only add results to initial exam }

end

else

begin

aExmRslts := TStringList.Create;

aExmRslts.Add(ADDTNLEXMRSLTS);

aExmRslts.Add(exmNme);

AddVendorRsltsToExmRslts(aExmRslts);

aExmRslts.Clear;

aExmRslts.Free;

end; { Add Pointer Exam Results to Memo for subsequent selected exams }

try

err := Not \_2507Info.CheckInExam(exmIEN, chkInDTM, fmMemoExmRslts);

except

on E: Exception do

begin

err := True; { So any previously modified exams are restored }

{ Handle Exception where the length of one or more lines in the Vendors

completed exam text file to be uploaded exceeds 999 characters in length. }

raise EVndrExmLngthError.Create('The length of one or more lines in the Vendors ASCII text file ' +

'is greater than 999 characters.' + #13#10 + 'Contact the Vendor to ' +

'have the file(s) corrected before trying to complete the check-in for this request again.');

end; { End Exception Handling }

end; { End Inner Try/Except Block }

until ((i = (exmObjLst.Count - 1)) or err);

{ No Errors and All Exams Completed/Cancelled - Set Request Status to 'P' }

if (Not err) and allExmsCmpltdCncld then

begin

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**\_2507Info.UpdtRqstStatus(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, '2') // Patch 193**

**else // Patch 193**

\_2507Info.UpdtRqstStatus(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, 'P'); // Patch 193

end;

finally

if err then { Error - restore modified exams to original state }

for i := 0 to exmObjLst.Count - 1 do

begin

exmIEN := exmObjLst[i];

{ Retrieve ORIGINAL exam results to restore }

fmMemoExmRslts.Lines.Assign(TExamInfo(exmObjLst.Objects[i]).GetExamResults);

\_2507Info.CheckInExam(exmIEN, '@', fmMemoExmRslts);

end; { End Modified Exam List Loop }

end; { End Outer Try/Finally Block }

Result := Not err;

end;

##### GUI – frmContractedExamDeleteCO.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| frmContractedExamDeleteCo.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events – N/A for Requirement 2.6.1

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| DeleteCheckOut | Function | Modified to handle Request Status database field change. |

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause

CapriSupport

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | DeleteCheckOut | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | DeleteCheckOut | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TExntdCntrctdBaseFormDeleteCO.DeleteCheckOut;

var

dlgType: TMsgDlgType;

err, updtRqstInfo: Boolean;

i: Integer;

exmIEN, msg, rqstVal: String;

begin

rqstVal := fmCmbBx2507Rqsts.Text;

msg := 'All Contracted Check-Out information has been successfully removed from the Exam(s).';

dlgType := mtConfirmation;

i := -1;

err := False;

updtRqstInfo := DelRqstInfo;

{ At Least One Record had to be selected to get here }

repeat

inc(i);

if chkdLstBxExams.ItemEnabled[i] then

if chkdLstBxExams.Checked[i] or chkBxCntrctAllExams.Checked then

begin

exmIEN := IntToStr(Integer(chkdLstBxExams.Items.Objects[i]));

if Not \_2507Info.CheckOutExam(exmIEN, '@', '@')then

err := True;

end; { End If Item enabled and is checked or all items checkbox is checked }

until (i = (chkdLstBxExams.Items.Count - 1));

{ Display Message to User }

if err then

begin

msg := 'An error occurred while removing the Contracted Check-Out information ' +

'from the selected exam entries.' + #13#10 + 'Reselect those available ' +

'exams on the request and try and delete the check-out again.' + #13#10#13#10 +

'If problems persist, please consult your local IRM for assistance.';

dlgType := mtError;

end { End If Error Occurred Deleting Check-Out Information }

else if updtRqstInfo then { Update Request Info if applicable if no errors }

begin

\_2507Info.UpdtRqstToSchld(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, 'P', '@')

end;

MessageDlgCAPRI(msg, dlgType, [mbOK], 0); { Display Appropriate Message to Screen }

{ If No errors and request info updated - refresh 2507 request info }

if ((Not err) and UpdtRqstInfo) then

begin

btnRfrshRqstsClick(self); { Refresh Request Info }

{ Reselect current selection }

fmCmbBx2507Rqsts.ItemIndex := fmCmbBx2507Rqsts.Items.IndexOf(rqstVal);

end;

end;

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

procedure TExntdCntrctdBaseFormDeleteCO.DeleteCheckOut;

var

dlgType: TMsgDlgType;

err, updtRqstInfo: Boolean;

i: Integer;

exmIEN, msg, rqstVal: String;

begin

rqstVal := fmCmbBx2507Rqsts.Text;

msg := 'All Contracted Check-Out information has been successfully removed from the Exam(s).';

dlgType := mtConfirmation;

i := -1;

err := False;

updtRqstInfo := DelRqstInfo;

{ At Least One Record had to be selected to get here }

repeat

inc(i);

if chkdLstBxExams.ItemEnabled[i] then

if chkdLstBxExams.Checked[i] or chkBxCntrctAllExams.Checked then

begin

exmIEN := IntToStr(Integer(chkdLstBxExams.Items.Objects[i]));

if Not \_2507Info.CheckOutExam(exmIEN, '@', '@')then

err := True;

end; { End If Item enabled and is checked or all items checkbox is checked }

until (i = (chkdLstBxExams.Items.Count - 1));

{ Display Message to User }

if err then

begin

msg := 'An error occurred while removing the Contracted Check-Out information ' +

'from the selected exam entries.' + #13#10 + 'Reselect those available ' +

'exams on the request and try and delete the check-out again.' + #13#10#13#10 +

'If problems persist, please consult your local IRM for assistance.';

dlgType := mtError;

end { End If Error Occurred Deleting Check-Out Information }

else if updtRqstInfo then { Update Request Info if applicable if no errors }

begin

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then**

\_2507Info.UpdtRqstToSchld(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, 'P', '@')

**else**

**\_2507Info.UpdtRqstToSchld(fmCmbBx2507Rqsts.GetSelectedRecord.IEN, '2', '@');**

end;

MessageDlgCAPRI(msg, dlgType, [mbOK], 0); { Display Appropriate Message to Screen }

{ If No errors and request info updated - refresh 2507 request info }

if ((Not err) and UpdtRqstInfo) then

begin

btnRfrshRqstsClick(self); { Refresh Request Info }

{ Reselect current selection }

fmCmbBx2507Rqsts.ItemIndex := fmCmbBx2507Rqsts.Items.IndexOf(rqstVal);

end;

end;

##### GUI – ManageReportsCCR.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| ManageReportsCCR.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| buttonReleaseClick | Button Click | Modified to handle Request Status database field change. |
| buttonReopenClick | Button Click | Modified to handle Request Status database field change. |

##### Methods – N/A for Requirement 2.6.1

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | buttonReleaseClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | buttonReleaseClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmManageReportsCCR.buttonReleaseClick(Sender: TObject);

var

nowdt: string;

whichIEN: string;

x: integer;

foundflag: boolean;

// ix: Integer;

// exResultID: string;

// Stn: string;

// Req: string;

// Exm: string;

// Idn: string;

// XMLRqstFile: string;

// CCRTempDir: string;

// xmlFileName: TFileName;

// rCon: string;

// localContractorIen: string;

begin

ResetWndwTmr;

foundflag := false;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if pos('[OPEN]', lstExamsRequested.Items[x]) > 0 then

foundflag := true;

if foundflag = true then begin

ShowMessageCAPRI('This exam request has open items and cannot be released yet.');

exit;

end;

if MessageDlgCAPRI('Are you ready to release this exam request so it can be printed by the requestor?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

{ When individual exams are [COMPLETED], i.e. right at fetch we don't tell VLER.

At the time of request being [RELEASED] we tell VLER for each of the exams that

has a VLER result value.}

{for each exam for whichIEN

PrepExamResultForFindCopyInsert(exResultID, 'Completed');

TransmitXMLMessageCompleteVLER

}

if lstExamsRequested.Items.Count > 0 then begin

if FMExamRequestListbox.ItemIndex > -1 then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

ReleaseVendorExamResultsToVLER(whichIEN);

end;

{ CCRTempDir := GetTempExmDir(FMGetsExamDetails.RPCBroker.User.DUZ); // CodeCR496 - LMS - 2014-07-14 consolidate/refactor all GetTempDir into one location, so all can benefit from Win 7 specific handling.

Stn := GetDivisionOfCBOC(FMGetsExamDetails.RPCBroker.User.Division); // DONE: LMS Piece(FMGetsExamDetails.RPCBroker.User.Division, '^');

for ix := 0 to lstExamsRequested.Items.Count - 1 do begin

FMGetsExamDetails.IENS := Piece(lstExamsRequested.Items[ix], '^', 1);

if FMGetsExamDetails.IENS = '' then

break;

FMGetsExamDetails.GetAndFill;

exResultID := FMGetsExamDetails.GetField('73').FMDBInternal; // Only set status complete for exam results that we've noted in this cell

if (exResultID <> '') and (exResultID <> CAPRI\_InhouseExamXMLNote) then begin // blank is exams that never got a vendor result and CAPRI\_InhouseExamXMLNote is for inhouse exams.

Req := FMGetsExamDetails.GetField('.02').FMDBInternal;

localContractorIen := FMGetsExamDetails.GetField('100').FMDBInternal;

rCon := TfrmCntrctdExmMain(Owner).GetRemoteConIen(localContractorIen); // LMS - New Optimized function. No need to go to DB if it's been already done.

// ExamName := Piece(lstExamsRequested.Items[ix], '^', 2);

PrepExamResultForFindCopyInsert(exResultID, 'Completed'); // this gets XMLResultFromVendor ready

Exm := Piece(lstExamsRequested.Items[ix], '^');

Idn := Stn + '-' + Req + '-' + Exm;

xmlFileName := idn + FormattedTimeStamp(Now()) + '.XML'; // DONE: LMS Make file unique to avoid collisions at Vendor

XMLRqstFile := CCRTempDir + '\' + xmlFileName;

XMLResultFromVendorNoContainer.SaveToFile(XMLRqstFile);

FormatXMLDoc(XMLRqstFile); // Add Line Breaks and Indents to make readable

TransmitXMLMessageCompleteVLER(XMLRqstFile, idn, rCon);

end;

end; }

end;

fCCOWBrkr.RemoteProcedure := 'ORWU DT';

fCCOWBrkr.Param[1].Value := 'NOW';

fCCOWBrkr.Param[1].PType := literal;

try

fCCOWBrkr.Call;

except

on EBrokerError do begin

Application.MessageBox('RPC ORWU DT could not be accessed!', 'Broker Error', MB\_OK);

exit;

end;

end;

nowdt := fCCOWBrkr.Results[0];

FMValidator1.SetUp('396.3', whichIEN, '12', NowDt); // Date Approved

FMValidator1.SetUp('396.3', whichIEN, '13', NowDt); // Date Released

FMValidator1.SetUp('396.3', whichIEN, '6', NowDt); // Date Completed

//frmMain.FMValidator1.SetUp('396.3',whichIEN,'14',AuthorIEN); // Released By

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'R'); // Req Status

if not FMValidator1.Validate then begin

FMValidator1.DisplayErrors;

exit;

end; {file}

FMfiler1.AddFDA('396.3', whichIEN, '12', NowDt); // Date Approved

FMfiler1.AddFDA('396.3', whichIEN, '13', NowDt); // Date Released

FMfiler1.AddFDA('396.3', whichIEN, '6', NowDt); // Date Completed

FMfiler1.AddFDA('396.3', whichIEN, '14', fCCOWBrkr.User.DUZ); // Released By

fmfiler1.AddFDA('396.3', whichIEN, '17', 'R'); // Request Status

if FMFiler1.AnythingToFile then

FMFiler1.Update;

// SendTIUNote; // Postponed to patch 190; rpk 9/24/2014

ExamRequestRefreshClick(Application);

end;

end; //

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

procedure TfrmManageReportsCCR.buttonReleaseClick(Sender: TObject);

var

nowdt: string;

whichIEN: string;

x: integer;

foundflag: boolean;

// ix: Integer;

// exResultID: string;

// Stn: string;

// Req: string;

// Exm: string;

// Idn: string;

// XMLRqstFile: string;

// CCRTempDir: string;

// xmlFileName: TFileName;

// rCon: string;

// localContractorIen: string;

begin

ResetWndwTmr;

foundflag := false;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if pos('[OPEN]', lstExamsRequested.Items[x]) > 0 then

foundflag := true;

if foundflag = true then begin

ShowMessageCAPRI('This exam request has open items and cannot be released yet.');

exit;

end;

if MessageDlgCAPRI('Are you ready to release this exam request so it can be printed by the requestor?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

{ When individual exams are [COMPLETED], i.e. right at fetch we don't tell VLER.

At the time of request being [RELEASED] we tell VLER for each of the exams that

has a VLER result value.}

{for each exam for whichIEN

PrepExamResultForFindCopyInsert(exResultID, 'Completed');

TransmitXMLMessageCompleteVLER

}

if lstExamsRequested.Items.Count > 0 then begin

if FMExamRequestListbox.ItemIndex > -1 then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

ReleaseVendorExamResultsToVLER(whichIEN);

end;

{ CCRTempDir := GetTempExmDir(FMGetsExamDetails.RPCBroker.User.DUZ); // CodeCR496 - LMS - 2014-07-14 consolidate/refactor all GetTempDir into one location, so all can benefit from Win 7 specific handling.

Stn := GetDivisionOfCBOC(FMGetsExamDetails.RPCBroker.User.Division); // DONE: LMS Piece(FMGetsExamDetails.RPCBroker.User.Division, '^');

for ix := 0 to lstExamsRequested.Items.Count - 1 do begin

FMGetsExamDetails.IENS := Piece(lstExamsRequested.Items[ix], '^', 1);

if FMGetsExamDetails.IENS = '' then

break;

FMGetsExamDetails.GetAndFill;

exResultID := FMGetsExamDetails.GetField('73').FMDBInternal; // Only set status complete for exam results that we've noted in this cell

if (exResultID <> '') and (exResultID <> CAPRI\_InhouseExamXMLNote) then begin // blank is exams that never got a vendor result and CAPRI\_InhouseExamXMLNote is for inhouse exams.

Req := FMGetsExamDetails.GetField('.02').FMDBInternal;

localContractorIen := FMGetsExamDetails.GetField('100').FMDBInternal;

rCon := TfrmCntrctdExmMain(Owner).GetRemoteConIen(localContractorIen); // LMS - New Optimized function. No need to go to DB if it's been already done.

// ExamName := Piece(lstExamsRequested.Items[ix], '^', 2);

PrepExamResultForFindCopyInsert(exResultID, 'Completed'); // this gets XMLResultFromVendor ready

Exm := Piece(lstExamsRequested.Items[ix], '^');

Idn := Stn + '-' + Req + '-' + Exm;

xmlFileName := idn + FormattedTimeStamp(Now()) + '.XML'; // DONE: LMS Make file unique to avoid collisions at Vendor

XMLRqstFile := CCRTempDir + '\' + xmlFileName;

XMLResultFromVendorNoContainer.SaveToFile(XMLRqstFile);

FormatXMLDoc(XMLRqstFile); // Add Line Breaks and Indents to make readable

TransmitXMLMessageCompleteVLER(XMLRqstFile, idn, rCon);

end;

end; }

end;

fCCOWBrkr.RemoteProcedure := 'ORWU DT';

fCCOWBrkr.Param[1].Value := 'NOW';

fCCOWBrkr.Param[1].PType := literal;

try

fCCOWBrkr.Call;

except

on EBrokerError do begin

Application.MessageBox('RPC ORWU DT could not be accessed!', 'Broker Error', MB\_OK);

exit;

end;

end;

nowdt := fCCOWBrkr.Results[0];

FMValidator1.SetUp('396.3', whichIEN, '12', NowDt); // Date Approved

FMValidator1.SetUp('396.3', whichIEN, '13', NowDt); // Date Released

FMValidator1.SetUp('396.3', whichIEN, '6', NowDt); // Date Completed

//frmMain.FMValidator1.SetUp('396.3',whichIEN,'14',AuthorIEN); // Released By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here. Do not validate IEN as that can return an error.**

**// frmMain.FMValidator1.Setup('396.3', whichIEN, '17', '4'); // Request Status IEN fails validation, comment out**

**if NOT IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'R'); // Req Status // Patch 193

if not FMValidator1.Validate then begin

FMValidator1.DisplayErrors;

exit;

end; {file}

FMfiler1.AddFDA('396.3', whichIEN, '12', NowDt); // Date Approved

FMfiler1.AddFDA('396.3', whichIEN, '13', NowDt); // Date Released

FMfiler1.AddFDA('396.3', whichIEN, '6', NowDt); // Date Completed

FMfiler1.AddFDA('396.3', whichIEN, '14', fCCOWBrkr.User.DUZ); // Released By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', '4') // Request Status // Patch 193**

**else // Patch 193**

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', 'R'); // Request Status // Patch 193

if FMFiler1.AnythingToFile then

FMFiler1.Update;

// SendTIUNote; // Postponed to patch 190; rpk 9/24/2014

ExamRequestRefreshClick(Application);

end;

end;

Table 48: Functions

| Function Name | buttonReopenClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | buttonReopenClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmManageReportsCCR.buttonReopenClick(Sender: TObject);

var

whichIEN: string;

begin

ResetWndwTmr;

if MessageDlgCAPRI('Re-opening will clear previous status information and may affect site statistics. Are you sure you want to re-open this exam?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

FMValidator1.SetUp('396.3', whichIEN, '6', ''); // Date Completed

FMValidator1.SetUp('396.3', whichIEN, '12', ''); // Date Approved

FMValidator1.SetUp('396.3', whichIEN, '13', ''); // Date Released

FMValidator1.SetUp('396.3', whichIEN, '14', ''); // Released By

FMValidator1.SetUp('396.3', whichIEN, '15', ''); // Date printed by the RO

FMValidator1.SetUp('396.3', whichIEN, '16', ''); // Printed By

FMValidator1.Setup('396.3', whichIEN, '17', 'S'); // Request Status

FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

FMValidator1.SetUp('396.3', whichIEN, '25', ''); // Approved By

FMValidator1.SetUp('396.3', whichIEN, '26', ''); // Approval Date/Time

if not FMValidator1.Validate then begin

FMValidator1.DisplayErrors;

exit;

end; {file}

FMfiler1.AddFDA('396.3', whichIEN, '6', ''); // Date Completed

FMfiler1.AddFDA('396.3', whichIEN, '12', ''); // Date Approved

FMfiler1.AddFDA('396.3', whichIEN, '13', ''); // Date Released

FMfiler1.AddFDA('396.3', whichIEN, '14', ''); // Released By

FMfiler1.AddFDA('396.3', whichIEN, '15', ''); // Date printed by the RO

FMfiler1.AddFDA('396.3', whichIEN, '16', ''); // Printed By

FMValidator1.Setup('396.3', whichIEN, '17', 'S'); // Request Status

FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

FMfiler1.AddFDA('396.3', whichIEN, '25', ''); // Approved By

FMfiler1.AddFDA('396.3', whichIEN, '26', ''); // Approval Date/Time

if FMFiler1.AnythingToFile then

FMFiler1.Update;

ExamRequestRefreshClick(Application);

end;

end;

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

procedure TfrmManageReportsCCR.buttonReopenClick(Sender: TObject);

var

whichIEN: string;

begin

ResetWndwTmr;

if MessageDlgCAPRI('Re-opening will clear previous status information and may affect site statistics. Are you sure you want to re-open this exam?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

FMValidator1.SetUp('396.3', whichIEN, '6', ''); // Date Completed

FMValidator1.SetUp('396.3', whichIEN, '12', ''); // Date Approved

FMValidator1.SetUp('396.3', whichIEN, '13', ''); // Date Released

FMValidator1.SetUp('396.3', whichIEN, '14', ''); // Released By

FMValidator1.SetUp('396.3', whichIEN, '15', ''); // Date printed by the RO

FMValidator1.SetUp('396.3', whichIEN, '16', ''); // Printed By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here. Do not validate IEN as that can return an error.**

**If NOT IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'S'); // Patch 193

FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

FMValidator1.SetUp('396.3', whichIEN, '25', ''); // Approved By

FMValidator1.SetUp('396.3', whichIEN, '26', ''); // Approval Date/Time

if not FMValidator1.Validate then begin

FMValidator1.DisplayErrors;

exit;

end; {file}

FMfiler1.AddFDA('396.3', whichIEN, '6', ''); // Date Completed

FMfiler1.AddFDA('396.3', whichIEN, '12', ''); // Date Approved

FMfiler1.AddFDA('396.3', whichIEN, '13', ''); // Date Released

FMfiler1.AddFDA('396.3', whichIEN, '14', ''); // Released By

FMfiler1.AddFDA('396.3', whichIEN, '15', ''); // Date printed by the RO

FMfiler1.AddFDA('396.3', whichIEN, '16', ''); // Printed By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here. Do not validate IEN as that can return an error.**

**if NOT IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

FMValidator1.Setup('396.3', whichIEN, '17', 'S'); // Patch 193

FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

FMfiler1.AddFDA('396.3', whichIEN, '25', ''); // Approved By

FMfiler1.AddFDA('396.3', whichIEN, '26', ''); // Approval Date/Time

if FMFiler1.AnythingToFile then

FMFiler1.Update;

ExamRequestRefreshClick(Application);

##### GUI – ManageReports.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| ManageReports.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| buttonReleaseClick | Button Click | Modified to handle Request Status database field change. |
| buttonReopenClick | Button Click | Modified to handle Request Status database field change. |

##### Methods – N/A for Requirement 2.6.1

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | buttonReleaseClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | buttonReleaseClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmManageReports.buttonReleaseClick(Sender: TObject);

var

nowdt: string;

whichIEN: string;

x: integer;

foundflag: boolean;

begin

foundflag := false;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if pos('[OPEN]', lstExamsRequested.Items[x]) > 0 then

foundflag := true;

if foundflag = true then begin

showmessageCAPRI('This exam request has open items and cannot be released yet.');

exit;

end;

if MessageDlgCAPRI('Are you ready to release this exam request so it can be printed by the requestor?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

if ANURemoteProcedureCallInProgress = True then

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

exit;

end;

end;

nowdt := frmMain.RPCBroker1.Results[0];

frmMain.FMValidator1.SetUp('396.3', whichIEN, '12', NowDt); // Date Approved

frmMain.FMValidator1.SetUp('396.3', whichIEN, '13', NowDt); // Date Released

frmMain.FMValidator1.SetUp('396.3', whichIEN, '6', NowDt); // Date Completed

//frmMain.FMValidator1.SetUp('396.3',whichIEN,'14',AuthorIEN); // Released By

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'R'); // Req Status

(\*

fmEditRequestStatus.Text:='R'; // Request Status

fmEditRequestStatus.IENS:=whichIEN;

if FMEditRequestStatus.FMModified then if FMEditRequestStatus.Validate then begin

frmMain.FMFiler1.AddChgdControl(FMEditRequestStatus);FMEditRequestStatus.Text:=FMEditRequestStatus.FMCtrlExternal;

end else begin

frmMain.FMValidator1.DisplayErrors; FMEditRequestStatus.Text:=FMEditRequestStatus.FMTag;

end;

\*)

if not frmMain.FMValidator1.Validate then begin

frmMain.FMValidator1.DisplayErrors;

exit;

end; {file}

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '12', NowDt); // Date Approved

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '13', NowDt); // Date Released

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '6', NowDt); // Date Completed

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '14', AuthorIEN); // Released By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', 'R'); // Request Status

if frmMain.FMFiler1.AnythingToFile then

begin

frmMain.FMFiler1.Update;

//CodeCR266 - rpm 1/18/12 - Send email notification

frmMain.MailManBuffer.Lines.Clear;

frmMain.MailManBuffer.Lines.Add('');

SendNotification(frmMain.MailManBuffer.Lines,

AuthorIEN,

'CAPRI: Release of 2507 Request',

'DVBA C 2507 EXAM READY',

whichIEN);

end;

ExamRequestRefreshClick(Application);

end;

end;

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

procedure TfrmManageReports.buttonReleaseClick(Sender: TObject);

var

nowdt: string;

whichIEN: string;

x: integer;

foundflag: boolean;

begin

foundflag := false;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if pos('[OPEN]', lstExamsRequested.Items[x]) > 0 then

foundflag := true;

if foundflag = true then begin

showmessageCAPRI('This exam request has open items and cannot be released yet.');

exit;

end;

if MessageDlgCAPRI('Are you ready to release this exam request so it can be printed by the requestor?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

if ANURemoteProcedureCallInProgress = True then

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

exit;

end;

end;

nowdt := frmMain.RPCBroker1.Results[0];

frmMain.FMValidator1.SetUp('396.3', whichIEN, '12', NowDt); // Date Approved

frmMain.FMValidator1.SetUp('396.3', whichIEN, '13', NowDt); // Date Released

frmMain.FMValidator1.SetUp('396.3', whichIEN, '6', NowDt); // Date Completed

//frmMain.FMValidator1.SetUp('396.3',whichIEN,'14',AuthorIEN); // Released By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here. Do not validate IEN, causes an error.**

**// frmMain.FMValidator1.Setup('396.3', whichIEN, '17', '4'); // Request Status IEN fails validation, comment out**

**if NOT IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'R'); // Req Status // Patch 193

(\*

fmEditRequestStatus.Text:='R'; // Request Status

fmEditRequestStatus.IENS:=whichIEN;

if FMEditRequestStatus.FMModified then if FMEditRequestStatus.Validate then begin

frmMain.FMFiler1.AddChgdControl(FMEditRequestStatus);FMEditRequestStatus.Text:=FMEditRequestStatus.FMCtrlExternal;

end else begin

frmMain.FMValidator1.DisplayErrors; FMEditRequestStatus.Text:=FMEditRequestStatus.FMTag;

end;

\*)

if not frmMain.FMValidator1.Validate then begin

frmMain.FMValidator1.DisplayErrors;

exit;

end; {file}

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '12', NowDt); // Date Approved

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '13', NowDt); // Date Released

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '6', NowDt); // Date Completed

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '14', AuthorIEN); // Released By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**if IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', '4') // Request Status // Patch 193**

**else** // Patch 193

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', 'R'); // Request Status // Patch 193

if frmMain.FMFiler1.AnythingToFile then

begin

frmMain.FMFiler1.Update;

//CodeCR266 - rpm 1/18/12 - Send email notification

frmMain.MailManBuffer.Lines.Clear;

frmMain.MailManBuffer.Lines.Add('');

SendNotification(frmMain.MailManBuffer.Lines,

AuthorIEN,

'CAPRI: Release of 2507 Request',

'DVBA C 2507 EXAM READY',

whichIEN);

end;

ExamRequestRefreshClick(Application);

end;

end;

Table 48: Functions

| Function Name | buttonReopenClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | buttonReopenClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmManageReports.buttonReopenClick(Sender: TObject);

var

whichIEN: string;

begin

if MessageDlgCAPRI('Re-opening will clear previous status information and may affect site statistics. Are you sure you want to re-open this exam?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

if ANURemoteProcedureCallInProgress = True then

exit;

frmMain.FMValidator1.SetUp('396.3', whichIEN, '6', ''); // Date Completed

frmMain.FMValidator1.SetUp('396.3', whichIEN, '12', ''); // Date Approved

frmMain.FMValidator1.SetUp('396.3', whichIEN, '13', ''); // Date Released

frmMain.FMValidator1.SetUp('396.3', whichIEN, '14', ''); // Released By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '15', ''); // Date printed by the RO

frmMain.FMValidator1.SetUp('396.3', whichIEN, '16', ''); // Printed By

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'O'); // Request Status

frmMain.FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

frmMain.FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '25', ''); // Approved By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '26', ''); // Approval Date/Time

(\*

fmEditRequestStatus.Text:='O'; // Request Status

fmEditRequestStatus.IENS:=whichIEN;

if FMEditRequestStatus.FMModified then if FMEditRequestStatus.Validate then begin

frmMain.fmFiler1.AddChgdControl(FMEditRequestStatus);FMEditRequestStatus.Text:=FMEditRequestStatus.FMCtrlExternal;

end else begin

frmMain.FMValidator1.DisplayErrors; FMEditRequestStatus.Text:=FMEditRequestStatus.FMTag;

end;

\*)

if not frmMain.FMValidator1.Validate then begin

frmMain.FMValidator1.DisplayErrors;

exit;

end; {file}

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '6', ''); // Date Completed

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '12', ''); // Date Approved

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '13', ''); // Date Released

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '14', ''); // Released By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '15', ''); // Date printed by the RO

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '16', ''); // Printed By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', 'O'); // Request Status

frmMain.FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

frmMain.FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '25', ''); // Approved By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '26', ''); // Approval Date/Time

if frmMain.FMFiler1.AnythingToFile then

frmMain.FMFiler1.Update;

ExamRequestRefreshClick(Application);

end;

end;

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

procedure TfrmManageReports.buttonReopenClick(Sender: TObject);

var

whichIEN: string;

begin

if MessageDlgCAPRI('Re-opening will clear previous status information and may affect site statistics. Are you sure you want to re-open this exam?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then begin

whichIEN := FMExamRequestListbox.GetSelectedRecord.IEN;

if ANURemoteProcedureCallInProgress = True then

exit;

frmMain.FMValidator1.SetUp('396.3', whichIEN, '6', ''); // Date Completed

frmMain.FMValidator1.SetUp('396.3', whichIEN, '12', ''); // Date Approved

frmMain.FMValidator1.SetUp('396.3', whichIEN, '13', ''); // Date Released

frmMain.FMValidator1.SetUp('396.3', whichIEN, '14', ''); // Released By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '15', ''); // Date printed by the RO

frmMain.FMValidator1.SetUp('396.3', whichIEN, '16', ''); // Printed By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**frmMain.FMValidator1.Setup('396.3', whichIEN, '17', '15') // Request Status // Patch 193**

**else // Patch 193**

frmMain.FMValidator1.Setup('396.3', whichIEN, '17', 'O'); // Request Status // Patch 193

frmMain.FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

frmMain.FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '25', ''); // Approved By

frmMain.FMValidator1.SetUp('396.3', whichIEN, '26', ''); // Approval Date/Time

(\*

fmEditRequestStatus.Text:='O'; // Request Status

fmEditRequestStatus.IENS:=whichIEN;

if FMEditRequestStatus.FMModified then if FMEditRequestStatus.Validate then begin

frmMain.fmFiler1.AddChgdControl(FMEditRequestStatus);FMEditRequestStatus.Text:=FMEditRequestStatus.FMCtrlExternal;

end else begin

frmMain.FMValidator1.DisplayErrors; FMEditRequestStatus.Text:=FMEditRequestStatus.FMTag;

end;

\*)

if not frmMain.FMValidator1.Validate then begin

frmMain.FMValidator1.DisplayErrors;

exit;

end; {file}

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '6', ''); // Date Completed

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '12', ''); // Date Approved

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '13', ''); // Date Released

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '14', ''); // Released By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '15', ''); // Date printed by the RO

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '16', ''); // Printed By

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', '15') // Request Status // Patch 193**

**else // Patch 193**

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '17', 'O'); // Request Status // Patch 193

frmMain.FMValidator1.Setup('396.3', whichIEN, '19', ''); // Cancellation Date

frmMain.FMValidator1.SetUp('396.3', whichIEN, '20', ''); // Cancelled By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '25', ''); // Approved By

frmMain.fmfiler1.AddFDA('396.3', whichIEN, '26', ''); // Approval Date/Time

if frmMain.FMFiler1.AnythingToFile then

frmMain.FMFiler1.Update;

ExamRequestRefreshClick(Application);

end;

end;

##### GUI – TIUSign.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| TIUSign.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events – N/A for Requirement 2.6.1

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| ReleaseExamToRO | Procedure | Modified to handle Request Status database field change. |

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | ReleaseExamToRO | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | ReleaseExamToRO | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TTIUSignForm.ReleaseExamToRO;

begin

// Do the release update here

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do

begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

end;

end;

// AMIE thinks Other Documents Required is a necessary field, but the RO doesn't

// fill it in, so this code fails if validated.

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]); //Date completed

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]); // Date released

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN); // Released by

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R'); // Released by

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]);

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]);

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN);

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R');

if FMFiler1.AnythingToFile then

if FMFiler1.DataProblemCheck then

FMFiler1.ProcessDataProblemList

else

begin

FMFiler1.Update;

// CodeCR471 JRL 4/26/13 - Remove email notification. This now happens as each

// exam is signed and not at the end of the Request

// //CodeCR266 - rpm 1/18/12 - Send email notification

// frmMain.MailManBuffer.Lines.Clear;

// frmMain.MailManBuffer.Lines.Add('');

// SendNotification(frmMain.MailManBuffer.Lines,

// AuthorIEN,

// 'CAPRI: Release of 2507 Request',

// 'DVBA C 2507 EXAM READY',

// FMExamRequestListbox.GetSelectedRecord.IEN);

ShowmessageCAPRI('The selected exam request has been released to the Regional Office.');

end;

end;

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

procedure TTIUSignForm.ReleaseExamToRO;

**var**

**Patch193Installed : Boolean;**

begin

**// Need to check the patch before calling RPC. If patch is checked during the**

**// routine, we lose the original RPC data**

**if IsPatchInstalled('DVBA\*2.7\*193') then**

**Patch193Installed := TRUE**

**else**

**Patch193Installed := FALSE;**

// Do the release update here

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do

begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

end;

end;

// AMIE thinks Other Documents Required is a necessary field, but the RO doesn't

// fill it in, so this code fails if validated.

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]); //Date completed

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]); // Date released

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN); // Released by

**// Request Status field has changed from a set of codes to a pointer. Update**

**// field with 'RELEASED TO RO, NOT PRINTED' IEN of 4. Patch 193 JRL 4/28/16**

**// Do not validate IEN here - may return an error.**

**if NOT Patch193Installed then**

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R'); // Released by

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]);

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]);

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN);

**// Request Status field has changed from a set of codes to a pointer. Updating**

**// field with 'RELEASED TO RO, NOT PRINTED' IEN of 4. Patch 193 JRL 4/28/16**

**if Patch193Installed then**

**FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', '4')**

**else**

FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R');

if FMFiler1.AnythingToFile then

if FMFiler1.DataProblemCheck then

FMFiler1.ProcessDataProblemList

else

begin

FMFiler1.Update;

// CodeCR471 JRL 4/26/13 - Remove email notification. This now happens as each

// exam is signed and not at the end of the Request

// //CodeCR266 - rpm 1/18/12 - Send email notification

// frmMain.MailManBuffer.Lines.Clear;

// frmMain.MailManBuffer.Lines.Add('');

// SendNotification(frmMain.MailManBuffer.Lines,

// AuthorIEN,

// 'CAPRI: Release of 2507 Request',

// 'DVBA C 2507 EXAM READY',

// FMExamRequestListbox.GetSelectedRecord.IEN);

ShowmessageCAPRI('The selected exam request has been released to the Regional Office.');

end;

end;

Table 48: Functions

| Function Name | SetMatchingTemplateComplete | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | SetMatchingTemplateComplete | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TTIUSignForm.SetMatchingTemplateComplete(aTIUNoteIEN: string);

var

I: Integer;

frmUncosigned: TfrmUncosigned;

begin

frmUncosigned := TfrmUncosigned.Create(TIUSignForm);

try

frmUncosigned.buttonsearchclick(application);

I := -1;

if frmUncosigned.FMListBoxTemplates.Items.Count > 0 then

repeat

inc(I);

if I > frmUncosigned.FMListBoxTemplates.Items.Count - 1 then

I := -1;

until (aTIUNoteIEN = findLastPiece(frmUncosigned.FMListBoxTemplates.Items[I])) or (I = -1) or (I = frmUncosigned.FMListBoxTemplates.Items.Count - 1);

if I > 0 then

begin

// Update CAPRI template status to complete

frmUncosigned.FMListBoxTemplates.ItemIndex := I;

frmMain.FMFiler1.AddFDA('396.17', frmUncosigned.FMListBoxTemplates.GetSelectedRecord.IEN, '11', 'C');

frmMain.FMFiler1.Update;

end

else

begin

ShowmessageCAPRI('NOTICE: Couldn''t find a matching template to close in CAPRI. The exam has been updated in AMIE, but no CAPRI templates have been modified.');

end;

finally

FreeAndNil(frmUncosigned);

end;

end;

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

procedure TTIUSignForm.SetMatchingTemplateComplete(aTIUNoteIEN: string);

var

I: Integer;

frmUncosigned: TfrmUncosigned;

begin

frmUncosigned := TfrmUncosigned.Create(TIUSignForm);

try

frmUncosigned.buttonsearchclick(application);

I := -1;

if frmUncosigned.FMListBoxTemplates.Items.Count > 0 then

repeat

inc(I);

if I > frmUncosigned.FMListBoxTemplates.Items.Count - 1 then

I := -1;

until (aTIUNoteIEN = findLastPiece(frmUncosigned.FMListBoxTemplates.Items[I])) or (I = -1) or (I = frmUncosigned.FMListBoxTemplates.Items.Count - 1);

if I > 0 then

begin

// Update CAPRI template status to complete

frmUncosigned.FMListBoxTemplates.ItemIndex := I;

**// Patch 193 JRL 4/29/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here.**

**If IsPatchInstalled('DVBA\*2.7\*193') then**

**frmMain.FMFiler1.AddFDA('396.17', frmUncosigned.FMListBoxTemplates.GetSelectedRecord.IEN, '11', '5')**

**else**

frmMain.FMFiler1.AddFDA('396.17', frmUncosigned.FMListBoxTemplates.GetSelectedRecord.IEN, '11', 'C');

frmMain.FMFiler1.Update;

end

else

begin

ShowmessageCAPRI('NOTICE: Couldn''t find a matching template to close in CAPRI. The exam has been updated in AMIE, but no CAPRI templates have been modified.');

end;

finally

FreeAndNil(frmUncosigned);

end;

end;

##### GUI – UncosignedUtility.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| UncosignedUtility.pas | Modified to handle Request Status database field change. |

##### GUI Classes – N/A for Requirement 2.6.1

##### Current Form – N/A for Requirement 2.6.1

##### Modified Form – N/A for Requirement 2.6.1

##### Components on Form – N/A for Requirement 2.6.1

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| Button3Click | Button Click | Modified to handle Request Status database field change. |

##### Methods – N/A for Requirement 2.6.1

##### Special References – N/A for Requirement 2.6.1

##### Class Events – N/A for Requirement 2.6.1

##### Class Methods – N/A for Requirement 2.6.1

##### Class Properties – N/A for Requirement 2.6.1

##### Uses Clause – N/A for Requirement 2.6.1

##### Forms – N/A for Requirement 2.6.1

##### Functions

Table 48: Functions

| Function Name | Button3Click | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modified to handle Request Status database field change. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | Button3Click | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmUncosigned.Button3Click(Sender: TObject);

var

x, zz: integer;

foundflag: boolean;

OriginalExamInfo: string;

OKToCloseFlag: Boolean;

TranscribedFound: Boolean;

begin

comboBoxExamLocation.ItemIndex := 0;

foundflag := false;

ComboBoxReleaseReport.ItemIndex := 0;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if lstExamsRequested.Selected[x] = True then

FoundFlag := True;

if foundflag = false then begin

ShowMessageCAPRI('Please select an exam to complete.');

exit;

end;

if FMExProvider.Text = '' then begin

ShowMessageCAPRI('Please enter an examining provider name.');

exit;

end;

if ComboBoxExamLocation.ItemIndex = -1 then begin

ShowMessageCAPRI('Please pick an exam location.');

exit;

end;

/////////////////////////////////////////////////////

OriginalExamInfo := '';

for zz := 0 to lstExamsRequested.Items.Count - 1 do begin

if lstExamsRequested.Selected[zz] = true then begin

FMMemoAMIEResults.IENS := Piece(lstExamsRequested.Items[zz], '^', 2);

if originalexaminfo = '' then begin

// Only put report in first exam type

QuickCopy(MemoDocs, FMMemoAMIEResults);

originalexaminfo := Piece(Piece(lstExamsRequested.Items[zz], '^', 1), '[', 1);

end

else begin

// Reference the saved report here

FMMemoAMIEResults.Lines.clear;

FMMemoAMIEResults.Lines.add('Please refer to the exam report contained in:');

FMMemoAMIEResults.Lines.add(originalexaminfo);

end;

FMExProvider.IENS := FMMemoAMIEResults.IENS;

if FMExProvider.FMModified then

if FMExProvider.Validate then begin

frmMain.FMFiler1.AddChgdControl(FMExProvider);

FMExProvider.Text := FMExProvider.FMCtrlExternal;

end

else begin

frmMain.FMValidator1.DisplayErrors;

FMExProvider.Text := FMExProvider.FMTag;

end;

frmMain.FMFiler1.AddChgdControl(FMMemoAMIEResults);

if ComboBoxReleaseReport.ItemIndex = 0 then begin

// Set Status

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '.04', 'C');

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '.04', 'C');

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '75', TIUNoteIEN);

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '75', TIUNoteIEN);

end;

// Set Location

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '.09', Copy(ComboBoxExamLocation.Text, 1, 1));

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '.09', Copy(ComboBoxExamLocation.Text, 1, 1));

(\*

// Fee Exam?

frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'.08',Copy(ComboBoxFeeExam.Text,1,1));

frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'.08',Copy(ComboBoxFeeExam.Text,1,1));

\*)

// Date of Exam

//frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'.06',FloatToStr(ORDateCombo1.FMDate));

//frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'.06',FloatToStr(ORDateCombo1.FMDate));

(\*

// Date Released

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall; try frmMain.RPCBroker1.Call;

except

On EBrokerError do begin; ANURemoteProcedureCallInProgress:=False;

ShowMessageCAPRI('ORWU DT could not be accessed!');

application.terminate;

end; end;

frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'13',frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'13',frmMain.RPCBroker1.Results[0]);

\*)

if frmMain.FMFiler1.AnythingToFile then

frmMain.FMFiler1.Update;

end;

(\*

if frmMain.FMFiler1.AnythingToFile then

if frmMain.FMFiler1.DataProblemCheck then

frmMain.FMFiler1.ProcessDataProblemList

else

frmMain.FMFiler1.Update;

end;

\*)

end;

// Do a check and see if it's ok to release the results to the RO

FMExamRequestListboxClick(Application);

// Be sure that nothing is still open

if lstExamsRequested.Items.Count = 0 then begin

// Check each exam result and be sure it's released prior

// to releasing the whole request

OKToCloseFlag := True;

for x := 0 to lstExamsRequestedHidden.Items.Count - 1 do begin

// Check if exam should be released but isn't

if (POS('[OPEN]', lstExamsRequestedHidden.Items[x]) > 0) then

if (POS('[TRANSFERRED OUT]', lstExamsRequestedHidden.Items[x]) > 0) then

OKToCloseFlag := False;

end;

// Check to be sure ALL existing exams do not have a date transcription complete

// value - file 396.4, field 90

// If any have a date, then we don't want to auto release

TranscribedFound := False;

if lstExamsRequestedHidden.Items.Count > 0 then

for x := 0 to lstExamsRequestedHidden.Items.Count - 1 do begin

FMGetsExamRequestInfo.IENS := Piece(lstExamsRequestedHidden.Items[x], '^', 1);

FMGetsExamRequestInfo.GetData;

//ShowmessageCAPRI(FMGetsExamRequestInfo.GetField('90').FMDInternal);

if FMGetsExamRequestInfo.GetField('90').FMDBExternal <> '' then

TranscribedFound := True;

end;

if (OKToCloseFlag = True) and (transcribedfound = false) then begin

// Do the release update here

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

end;

end;

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]); //Date completed

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]); // Date released

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN); // Released by

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R');

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN);

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R');

if frmMain.FMFiler1.AnythingToFile then

begin

frmMain.FMFiler1.Update;

//CodeCR266 - rpm 1/18/12 - Send email notification

frmMain.MailManBuffer.Lines.Clear;

frmMain.MailManBuffer.Lines.Add('');

SendNotification(frmMain.MailManBuffer.Lines,

AuthorIEN,

'CAPRI: Release of 2507 Request',

'DVBA C 2507 EXAM READY',

FMExamRequestListbox.GetSelectedRecord.IEN);

end;

end

else begin

ShowMessageCAPRI('Case not ready to be closed.');

end;

if TranscribedFound = True then begin

ShowmessageCAPRI('WARNING: This exam request appears to contain one or more transcribed exams. CAPRI will be unable to release it. Please use AMIE to review and release this exam request.');

end;

end;

frmMain.RPCBrokerCall;

frmMain.FMFiler1.AddFDA('396.17', FMListBoxTemplates.GetSelectedRecord.IEN, '11', 'C');

frmMain.RPCBrokerCall;

sleep(500);

fmfile;

/////////////////////////////////////////////////////

ContinueCosSearch := True;

PanelNoteText.Visible := False;

end;

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

procedure TfrmUncosigned.Button3Click(Sender: TObject);

var

x, zz: integer;

foundflag: boolean;

OriginalExamInfo: string;

OKToCloseFlag: Boolean;

TranscribedFound: Boolean;

begin

comboBoxExamLocation.ItemIndex := 0;

foundflag := false;

ComboBoxReleaseReport.ItemIndex := 0;

if lstExamsRequested.Items.Count > 0 then

for x := 0 to lstExamsRequested.Items.Count - 1 do

if lstExamsRequested.Selected[x] = True then

FoundFlag := True;

if foundflag = false then begin

ShowMessageCAPRI('Please select an exam to complete.');

exit;

end;

if FMExProvider.Text = '' then begin

ShowMessageCAPRI('Please enter an examining provider name.');

exit;

end;

if ComboBoxExamLocation.ItemIndex = -1 then begin

ShowMessageCAPRI('Please pick an exam location.');

exit;

end;

/////////////////////////////////////////////////////

OriginalExamInfo := '';

for zz := 0 to lstExamsRequested.Items.Count - 1 do begin

if lstExamsRequested.Selected[zz] = true then begin

FMMemoAMIEResults.IENS := Piece(lstExamsRequested.Items[zz], '^', 2);

if originalexaminfo = '' then begin

// Only put report in first exam type

QuickCopy(MemoDocs, FMMemoAMIEResults);

originalexaminfo := Piece(Piece(lstExamsRequested.Items[zz], '^', 1), '[', 1);

end

else begin

// Reference the saved report here

FMMemoAMIEResults.Lines.clear;

FMMemoAMIEResults.Lines.add('Please refer to the exam report contained in:');

FMMemoAMIEResults.Lines.add(originalexaminfo);

end;

FMExProvider.IENS := FMMemoAMIEResults.IENS;

if FMExProvider.FMModified then

if FMExProvider.Validate then begin

frmMain.FMFiler1.AddChgdControl(FMExProvider);

FMExProvider.Text := FMExProvider.FMCtrlExternal;

end

else begin

frmMain.FMValidator1.DisplayErrors;

FMExProvider.Text := FMExProvider.FMTag;

end;

frmMain.FMFiler1.AddChgdControl(FMMemoAMIEResults);

if ComboBoxReleaseReport.ItemIndex = 0 then begin

// Set Status

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '.04', 'C');

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '.04', 'C');

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '75', TIUNoteIEN);

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '75', TIUNoteIEN);

end;

// Set Location

frmMain.FMValidator1.SetUp('396.4', FMMemoAMIEResults.IENS, '.09', Copy(ComboBoxExamLocation.Text, 1, 1));

frmMain.FMFiler1.AddFDA('396.4', FMMemoAMIEResults.IENS, '.09', Copy(ComboBoxExamLocation.Text, 1, 1));

(\*

// Fee Exam?

frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'.08',Copy(ComboBoxFeeExam.Text,1,1));

frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'.08',Copy(ComboBoxFeeExam.Text,1,1));

\*)

// Date of Exam

//frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'.06',FloatToStr(ORDateCombo1.FMDate));

//frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'.06',FloatToStr(ORDateCombo1.FMDate));

(\*

// Date Released

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall; try frmMain.RPCBroker1.Call;

except

On EBrokerError do begin; ANURemoteProcedureCallInProgress:=False;

ShowMessageCAPRI('ORWU DT could not be accessed!');

application.terminate;

end; end;

frmMain.FMValidator1.SetUp('396.4',FMMemoAMIEResults.IENS,'13',frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.4',FMMemoAMIEResults.IENS,'13',frmMain.RPCBroker1.Results[0]);

\*)

if frmMain.FMFiler1.AnythingToFile then

frmMain.FMFiler1.Update;

end;

(\*

if frmMain.FMFiler1.AnythingToFile then

if frmMain.FMFiler1.DataProblemCheck then

frmMain.FMFiler1.ProcessDataProblemList

else

frmMain.FMFiler1.Update;

end;

\*)

end;

// Do a check and see if it's ok to release the results to the RO

FMExamRequestListboxClick(Application);

// Be sure that nothing is still open

if lstExamsRequested.Items.Count = 0 then begin

// Check each exam result and be sure it's released prior

// to releasing the whole request

OKToCloseFlag := True;

for x := 0 to lstExamsRequestedHidden.Items.Count - 1 do begin

// Check if exam should be released but isn't

if (POS('[OPEN]', lstExamsRequestedHidden.Items[x]) > 0) then

if (POS('[TRANSFERRED OUT]', lstExamsRequestedHidden.Items[x]) > 0) then

OKToCloseFlag := False;

end;

// Check to be sure ALL existing exams do not have a date transcription complete

// value - file 396.4, field 90

// If any have a date, then we don't want to auto release

TranscribedFound := False;

if lstExamsRequestedHidden.Items.Count > 0 then

for x := 0 to lstExamsRequestedHidden.Items.Count - 1 do begin

FMGetsExamRequestInfo.IENS := Piece(lstExamsRequestedHidden.Items[x], '^', 1);

FMGetsExamRequestInfo.GetData;

//ShowmessageCAPRI(FMGetsExamRequestInfo.GetField('90').FMDInternal);

if FMGetsExamRequestInfo.GetField('90').FMDBExternal <> '' then

TranscribedFound := True;

end;

if (OKToCloseFlag = True) and (transcribedfound = false) then begin

// Do the release update here

frmMain.RPCBroker1.RemoteProcedure := 'ORWU DT';

frmMain.RPCBroker1.Param[1].Value := 'NOW';

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBrokerCall;

try

frmMain.RPCBroker1.Call;

except

on EBrokerError do begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWU DT could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWU DT could not be accessed!');

end;

end;

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]); //Date completed

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]); // Date released

frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN); // Released by

**// Patch 193 JRL 5/5/16**

**// Changed Request Status field to pointer field. Using IEN to populate**

**// status here. Don't validate IEN, it will fail.**

**// frmMain.FMValidator1.Setup('396.3', whichIEN, '17', '4'); // Request Status IEN fails validation, comment out**

**if NOT IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193**

**frmMain.FMValidator1.SetUp('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R'); // Patch 193**

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '6', frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '13', frmMain.RPCBroker1.Results[0]);

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '14', AuthorIEN);

**// Patch 193 JRL 5/5/16**

**// Changed Request Status field to pointer field. Using IEN to populate status here.**

**if IsPatchInstalled('DVBA\*2.7\*193') then**

**frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', '4')**

**else**

frmMain.FMFiler1.AddFDA('396.3', FMExamRequestListbox.GetSelectedRecord.IEN, '17', 'R');

if frmMain.FMFiler1.AnythingToFile then

begin

frmMain.FMFiler1.Update;

//CodeCR266 - rpm 1/18/12 - Send email notification

frmMain.MailManBuffer.Lines.Clear;

frmMain.MailManBuffer.Lines.Add('');

SendNotification(frmMain.MailManBuffer.Lines,

AuthorIEN,

'CAPRI: Release of 2507 Request',

'DVBA C 2507 EXAM READY',

FMExamRequestListbox.GetSelectedRecord.IEN);

end;

end

else begin

ShowMessageCAPRI('Case not ready to be closed.');

end;

if TranscribedFound = True then begin

ShowmessageCAPRI('WARNING: This exam request appears to contain one or more transcribed exams. CAPRI will be unable to release it. Please use AMIE to review and release this exam request.');

end;

end;

frmMain.RPCBrokerCall;

frmMain.FMFiler1.AddFDA('396.17', FMListBoxTemplates.GetSelectedRecord.IEN, '11', 'C');

frmMain.RPCBrokerCall;

sleep(500);

fmfile;

/////////////////////////////////////////////////////

ContinueCosSearch := True;

PanelNoteText.Visible := False;

end;

##### Dialog – Not Applicable for Requirement 2.6.1

##### Help Frame – Not Applicable for Requirement 2.6.1

##### HL7 Application Parameter – Not Applicable for Requirement 2.6.1

##### HL7 Logical Link – Not Applicable for Requirement 2.6.1

##### COTS Interface – Not Applicable for Requirement 2.6.1

#### Requirement 2.6.2

CAPRI shall be modified so that when new routing locations are added to VistA the newly routed locations in CAPRI will by default be set to **inactive**. **(Business Requirements 608794, CR 96)**

#### Database Repository – N/A for Requirement 2.6.2.

#### System Features

* As a CAPRI user, I want new routing locations that are added to VistA to no longer **default to active** in CAPRI.
* As a CAPRI user, I want new routing locations added to VistA that is picked up by CAPRI to automatically **default to inactive** in CAPRI.
* As a CAPRI user, I want an option provided that will give me the ability to manually activate in CAPRI new routing locations that have been added from VistA and defaulted to inactive. This is currently managed within “edit exam list parameters-MAS”.

#### Design Element Tables

##### Routines (Entry Points) – N/A for Requirement 2.6.2

##### Templates – N/A for Requirement 2.6.2

##### Bulletins – N/A for Requirement 2.6.2

##### Data Entries Affected by the Design – N/A for Requirement 2.6.2

##### Unique Record(s) - N/A for Requirement 2.6.2

##### File or Global Size Changes – N/A for Requirement 2.6.2

##### Mail Groups – N/A for Requirement 2.6.2

##### Security Keys – N/A for Requirement 2.6.2

##### Options – N/A for Requirement 2.6.2

##### Protocols – N/A for Requirement 2.6.2

##### Remote Procedure Call (RPC) – N/A for Requirement 2.6.2

##### Constants Defined in Interface – N/A for Requirement 2.6.2

##### Variables Defined in Interface – N/A for Requirement 2.6.2

##### Types Defined in Interface – N/A for Requirement 2.6.2

##### GUI – Main.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| Main.pas | Alter menu event code for Edit Exam List Parameters (MAS). |

##### GUI Classes – N/A for Requirement 2.6.3

##### Current Form – N/A for Requirement 2.6.2

##### Modified Form – N/A for Requirement 2.6.2

##### Components on Form – N/A for Requirement 2.6.2

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| actToolsExamListParametersExecute | Event | Menu event for Edit Exam List Parameters (MAS) |

##### Methods

##### Special References – N/A for Requirement 2.6.3

##### Class Events – N/A for Requirement 2.6.3

##### Class Methods – N/A for Requirement 2.6.2

##### Class Properties – N/A for Requirement 2.6.2

##### Uses Clause – N/A for Requirement 2.6.2

##### Forms – N/A for Requirement 2.6.2

##### Functions

Table 48: Function

| Function Name | Activities | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Modify event. When it locates a new CAPRI division file entry, make that entry “Inactive” instead of “Active”. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name:N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

procedure TfrmMain.actToolsExamListParametersExecute(Sender: TObject);

Var

x, y, z: integer;

xx, yy, zz: integer;

foundflag: boolean;

tempSites: TStringList;

IEN, ZIEN: String;

Begin

If ANURemoteProcedureCallInProgress = True Then

exit;

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating exam assignment dialog.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

tempSites := TStringList.Create;

tempSites.Clear;

frmSiteExamList := TfrmSiteExamList.Create(frmMain);

//Set up exams list

frmSiteExamList.FMExamsList.Items := ListExams;

If frmSiteExamList.FMExamsList.Items.Count > 0 Then

For x := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

If Pos(' I', frmSiteExamList.FMExamsList.Items[x]) > 0 Then

frmSiteExamList.FMExamsList.Items.Delete(x);

// Make sure CAPRI DIVISION EXAM LIST FILE contains an entry in .01 for each division

// The site may have added a new division.

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := FMListerMedicalCenterDivision;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count = 0 Then

Begin

ShowMessageCAPRI('No divisions can be located in VistA. Exiting this function.');

frmSiteExamList.Release;

frmSiteExamList := Nil;

TempHSMemo.Free;

exit;

End;

// Get current list of divisions

For x := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

tempSites.Add(frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN);

End;

// Now get list of divisions defined in the CAPRI DIVISION EXAM LIST FILE

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := frmSiteExamList.FMListerCustomExamListDivisions;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Find IEN and show that in the list

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

frmSiteExamList.CustomExamListDivisions.IENS := frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmSiteExamList.CustomExamListDivisions.GetData;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] := frmSiteExamList.CustomExamListDivisions.GetField('.01').FMDBInternal;

End;

// Compare the two lists and add medical center divisions that are missing

For x := 0 To tempSites.Count - 1 Do

Begin

foundflag := false;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

If foundflag = false Then

Begin

If tempSites[x] = frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] Then

Begin

foundflag := true;

End;

End;

End;

If FoundFlag = False Then

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := '`' + tempSites[x];

frmSiteExamList.FMEditCAPRIDivisionExamList.IENS := '+1,';

If frmSiteExamList.FMEditCAPRIDivisionExamList.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIDivisionExamList);

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMTag;

exit;

End;

If Not frmMain.FMValidator1.Validate Then

Begin

FMFiler1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating new divisions.');

exit;

End;

IEN := FMFiler1.FindIEN('+1,') + ',';

FMFiler1.AddFDA('396.151', '+1,' + IEN, '.01', 'DEFAULT LIST');

FMFiler1.AddFDA('396.151', '+1,' + IEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

ZIEN := FMFiler1.FindIEN('+1,') + ',' + IEN;

For z := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x] + ', Exam: ' + frmSiteExamList.FMExamsList.Items[z];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMExamsList.ItemIndex := z;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := '`' + frmSiteExamList.FMExamsList.GetSelectedRecord.IEN;

frmSiteExamList.FMEditCAPRIExamListDefinition.IENS := '+1,' + ZIEN;

If frmSiteExamList.FMEditCAPRIExamListDefinition.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIExamListDefinition);

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMTag;

exit;

End;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

If Not frmMain.FMValidator1.Validate Then

Begin

FMValidator1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating the exam list.');

exit;

End;

End; // end of exam list population

End;

End;

//Reload division list in case it changed

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Now evaluate for exams that have been inactivated or exams that have been added

StatusBarLoadPt.Caption := 'Validating list consistency.';

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

// Get list for division

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

// Load up the list and see what exams are in it.

If frmSiteExamList.FMListBoxAvailableExamLists.Items.Count > 0 Then

frmSiteExamList.FMListerExamSetup.ListerFlags := [lfInternal];

For z := 0 To frmSiteExamList.FMListBoxAvailableExamLists.Items.Count - 1 Do

Begin

frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex := z;

StatusBarLoadPt.Caption := 'Validating ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMListBoxAvailableExamLists.Items[frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex];

frmSiteExamList.buttonEditClick(Application);

// Check for exams now inactive

For xx := frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMListBoxExamSetup.ItemIndex := xx;

foundflag := false;

For yy := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMExamsList.ItemIndex := yy;

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[xx], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[xx]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

StatusBarLoadPt.Caption := 'Removing ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.CheckListBoxExamList.Items[xx];

frmSiteExamList.FMEditExamNameInList.Text := '@';

frmSiteExamList.FMEditExamNameInList.IENS := frmSiteExamList.FMListBoxExamSetup.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' +

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

If frmSiteExamList.FMEditExamNameInList.Validate Then

Begin

frmMain.FMFiler1.AddChgdControl(frmSiteExamList.FMEditExamNameInList);

frmSiteExamList.FMEditExamNameInList.Text := frmSiteExamList.FMEditExamNameInList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

exit;

End;

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

For xx := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

foundflag := false;

frmSiteExamList.FMExamsList.ItemIndex := xx;

For yy := 0 To frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Do

Begin

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[yy], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[yy]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

//ShowMessageCAPRI('turn on '+frmSiteExamList.FMExamsList.Items[xx]);

StatusBarLoadPt.Caption := 'Adding ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMExamsList.Items[xx];

ZIEN := frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '.01', frmSiteExamList.FMExamsList.GetSelectedRecord.IEN);

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

End;

frmSiteExamList.FMListerExamSetup.ListerFlags := [];

End;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := 0;

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

frmSiteExamList.top := frmMain.Top + ((frmMain.Height - frmSiteExamList.Height) Div 2);

frmSiteExamList.left := frmMain.left + ((frmMain.Width - frmSiteExamList.Width) Div 2);

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.Panel1.BringToFront;

// Set Up Institutions

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmSiteExamList.FMListBoxUserGroup.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

frmSiteExamList.FMMemoDivisionComments.Enabled := True;

ContextorChangeMessage := 'You are editing site parameters. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

frmSiteExamList.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmSiteExamList.Release;

frmSiteExamList := Nil;

tempSites.Free;

end;

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

procedure TfrmMain.actToolsExamListParametersExecute(Sender: TObject);

Var

x, y, z: integer;

xx, yy, zz: integer;

foundflag: boolean;

tempSites: TStringList;

IEN, ZIEN: String;

Begin

If ANURemoteProcedureCallInProgress = True Then

exit;

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating exam assignment dialog.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

tempSites := TStringList.Create;

tempSites.Clear;

frmSiteExamList := TfrmSiteExamList.Create(frmMain);

//Set up exams list

frmSiteExamList.FMExamsList.Items := ListExams;

If frmSiteExamList.FMExamsList.Items.Count > 0 Then

For x := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

If Pos(' I', frmSiteExamList.FMExamsList.Items[x]) > 0 Then

frmSiteExamList.FMExamsList.Items.Delete(x);

// Make sure CAPRI DIVISION EXAM LIST FILE contains an entry in .01 for each division

// The site may have added a new division.

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := FMListerMedicalCenterDivision;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count = 0 Then

Begin

ShowMessageCAPRI('No divisions can be located in VistA. Exiting this function.');

frmSiteExamList.Release;

frmSiteExamList := Nil;

TempHSMemo.Free;

exit;

End;

// Get current list of divisions

For x := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := x;

tempSites.Add(frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN);

End;

// Now get list of divisions defined in the CAPRI DIVISION EXAM LIST FILE

frmSiteExamList.FMListerCustomExamListDivisionsListBox.FMLister := frmSiteExamList.FMListerCustomExamListDivisions;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Find IEN and show that in the list

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

frmSiteExamList.CustomExamListDivisions.IENS := frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

frmSiteExamList.CustomExamListDivisions.GetData;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] := frmSiteExamList.CustomExamListDivisions.GetField('.01').FMDBInternal;

End;

// Compare the two lists and add medical center divisions that are missing

For x := 0 To tempSites.Count - 1 Do

Begin

foundflag := false;

If frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count > 0 Then

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

If foundflag = false Then

Begin

If tempSites[x] = frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[y] Then

Begin

foundflag := true;

End;

End;

End;

If FoundFlag = False Then

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := '`' + tempSites[x];

frmSiteExamList.FMEditCAPRIDivisionExamList.IENS := '+1,';

If frmSiteExamList.FMEditCAPRIDivisionExamList.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIDivisionExamList);

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIDivisionExamList.Text := frmSiteExamList.FMEditCAPRIDivisionExamList.FMTag;

exit;

End;

If Not frmMain.FMValidator1.Validate Then

Begin

FMFiler1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating new divisions.');

exit;

End;

IEN := FMFiler1.FindIEN('+1,') + ',';

**// Patch 193 - New divisions should be set to inactive instead of active.**

**// This keeps VBA users from erroneously sending requests to new divisions**

**// that are not prepared to receive them. Initially Remedy INC000001290246**

**// but not a defect. Incorporated into Patch 193 requirements.**

**// Patch 193 RRA 7/18/16**

**If IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 RRA 7/18/16**

**begin // Patch 193 RRA 7/18/16**

**FMFiler1.AddFDA('396.15', IEN, '4', 'Y'); // Patch 193 RRA 7/18/16**

**if not FMFiler1.Update then // Patch 193 RRA 7/18/16**

**begin // Patch 193 RRA 7/18/16**

**ShowMessageCAPRI('There was a problem updting new divisions.'); // Patch 193 RRA 7/18/16**

**exit; // Patch 193 RRA 7/18/16**

**end; // Patch 193 RRA 7/18/16**

**end; // Patch 193 RRA 7/18/16**

**// end RRA**

FMFiler1.AddFDA('396.151', '+1,' + IEN, '.01', 'DEFAULT LIST');

FMFiler1.AddFDA('396.151', '+1,' + IEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

ZIEN := FMFiler1.FindIEN('+1,') + ',' + IEN;

For z := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

StatusBarLoadPt.Caption := 'Setting up new division: IEN ' + tempSites[x] + ', Exam: ' + frmSiteExamList.FMExamsList.Items[z];

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.FMExamsList.ItemIndex := z;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := '`' + frmSiteExamList.FMExamsList.GetSelectedRecord.IEN;

frmSiteExamList.FMEditCAPRIExamListDefinition.IENS := '+1,' + ZIEN;

If frmSiteExamList.FMEditCAPRIExamListDefinition.Validate Then

Begin

FMFiler1.AddChgdControl(frmSiteExamList.FMEditCAPRIExamListDefinition);

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

frmSiteExamList.FMEditCAPRIExamListDefinition.Text := frmSiteExamList.FMEditCAPRIExamListDefinition.FMTag;

exit;

End;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

If Not frmMain.FMValidator1.Validate Then

Begin

FMValidator1.DisplayErrors;

exit;

End

Else

If Not FMFiler1.Update Then

Begin

ShowMessageCAPRI('There was a problem populating the exam list.');

exit;

End;

End; // end of exam list population

End;

End;

//Reload division list in case it changed

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetList;

// Now evaluate for exams that have been inactivated or exams that have been added

StatusBarLoadPt.Caption := 'Validating list consistency.';

For y := 0 To frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items.Count - 1 Do

Begin

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := y;

// Get list for division

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

// Load up the list and see what exams are in it.

If frmSiteExamList.FMListBoxAvailableExamLists.Items.Count > 0 Then

frmSiteExamList.FMListerExamSetup.ListerFlags := [lfInternal];

For z := 0 To frmSiteExamList.FMListBoxAvailableExamLists.Items.Count - 1 Do

Begin

frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex := z;

StatusBarLoadPt.Caption := 'Validating ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMListBoxAvailableExamLists.Items[frmSiteExamList.FMListBoxAvailableExamLists.ItemIndex];

frmSiteExamList.buttonEditClick(Application);

// Check for exams now inactive

For xx := frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMListBoxExamSetup.ItemIndex := xx;

foundflag := false;

For yy := frmSiteExamList.FMExamsList.Items.Count - 1 Downto 0 Do

Begin

frmSiteExamList.FMExamsList.ItemIndex := yy;

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[xx], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[xx]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

StatusBarLoadPt.Caption := 'Removing ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.CheckListBoxExamList.Items[xx];

frmSiteExamList.FMEditExamNameInList.Text := '@';

frmSiteExamList.FMEditExamNameInList.IENS := frmSiteExamList.FMListBoxExamSetup.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' +

frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

If frmSiteExamList.FMEditExamNameInList.Validate Then

Begin

frmMain.FMFiler1.AddChgdControl(frmSiteExamList.FMEditExamNameInList);

frmSiteExamList.FMEditExamNameInList.Text := frmSiteExamList.FMEditExamNameInList.FMCtrlExternal;

End

Else

Begin

frmMain.FMValidator1.DisplayErrors;

exit;

End;

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

For xx := 0 To frmSiteExamList.FMExamsList.Items.Count - 1 Do

Begin

foundflag := false;

frmSiteExamList.FMExamsList.ItemIndex := xx;

For yy := 0 To frmSiteExamList.FMListBoxExamSetup.Items.Count - 1 Do

Begin

If frmSiteExamList.FMExamsList.GetSelectedRecord.IEN = Copy(frmSiteExamList.FMListBoxExamSetup.Items[yy], 1, Pos(' ', frmSiteExamList.FMListBoxExamSetup.Items[yy]) - 1) Then

foundflag := true;

End;

If foundflag = false Then

Begin

//ShowMessageCAPRI('turn on '+frmSiteExamList.FMExamsList.Items[xx]);

StatusBarLoadPt.Caption := 'Adding ' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.Items[frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex] + ': ' +

frmSiteExamList.FMExamsList.Items[xx];

ZIEN := frmSiteExamList.FMListBoxAvailableExamLists.GetSelectedRecord.IEN + ',' + frmSiteExamList.FMListerCustomExamListDivisionsListBox.GetSelectedRecord.IEN;

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '.01', frmSiteExamList.FMExamsList.GetSelectedRecord.IEN);

FMFiler1.AddFDA('396.1514', '+1,' + ZIEN, '2', 'Y');

For zz := 1 To 50 Do

Begin

sleep(1); // Need this to deal with Broker calls clicked quickly in succession.

application.processmessages;

End;

//fmfile;

If Not frmMain.FMFiler1.Update Then

Begin

ShowmessageCAPRI('Problem updating exam specs. Please try again.');

application.terminate;

application.terminate;

End;

End;

End;

End;

frmSiteExamList.FMListerExamSetup.ListerFlags := [];

End;

frmSiteExamList.FMListerCustomExamListDivisionsListBox.ItemIndex := 0;

frmSiteExamList.FMListerCustomExamListDivisionsListBoxClick(Application);

frmSiteExamList.top := frmMain.Top + ((frmMain.Height - frmSiteExamList.Height) Div 2);

frmSiteExamList.left := frmMain.left + ((frmMain.Width - frmSiteExamList.Width) Div 2);

AnimateLogo(False);

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

frmSiteExamList.Panel1.BringToFront;

// Set Up Institutions

If frmROFinder.ORListBoxSites.Items.Count = 0 Then

LoadInstitutions;

(\*

If lstInstitutions.Items.Count>0 then

For x:=0 to lstInstitutions.Items.Count-1 do

frmSiteExamList.FMListBoxUserGroup.Items.Add(Copy(lstInstitutions.Items[x],1,Pos('^',lstInstitutions.Items[x])-1));

\*)

frmSiteExamList.FMMemoDivisionComments.Enabled := True;

ContextorChangeMessage := 'You are editing site parameters. If you continue, CAPRI will drop out of the clinical context.';

CCOWBreakLink := True;

frmSiteExamList.ShowModal;

ContextorChangeMessage := '';

CCOWBreakLink := False;

frmSiteExamList.Release;

frmSiteExamList := Nil;

tempSites.Free;

end;

##### Dialog – N/A for Requirement 2.6.2

##### Help Frame – N/A for Requirement 2.6.2

##### HL7 Application Parameter – N/A for Requirement 2.6.2

##### HL7 Logical Link – N/A for Requirement 2.6.2

##### COTS Interface – N/A for Requirement 2.6.2

#### Requirement 2.6.3

#### CAPRI shall be modified during the BHIE transition to have the DoD tab disabled within CAPRI. The business will notify the development team when the disabling should occur. (Business Requirements 608696)Database Repository – N/A for Requirement 2.6.3

#### Database Repository – N/A for Requirement 2.6.3.

#### System Features

* As a CAPRI user, I want CAPRI to no longer display the DoD tab.
* As a CAPRI user, I want the link to the VistA Web tab active. (Existing functionality)
* As a CAPRI user, I want the patient selection screen where the display of DoD information is located to be removed. (i.e., station 200 details will no longer exist)

#### Design Element Tables

##### Routines (Entry Points) – N/A for Requirement 2.6.2

##### Templates – N/A for Requirement 2.6.2

##### Bulletins – N/A for Requirement 2.6.2

##### Data Entries Affected by the Design – N/A for Requirement 2.6.2

##### Unique Record(s) – N/A for Requirement 2.6.2

##### File or Global Size Changes – N/A for Requirement 2.6.2

##### Mail Groups – N/A for Requirement 2.6.2

##### Security Keys – N/A for Requirement 2.6.2

##### Options – N/A for Requirement 2.6.2

##### Protocols – N/A for Requirement 2.6.2

##### Remote Procedure Call (RPC) – N/A for Requirement 2.6.2

##### Constants Defined in Interface – N/A for Requirement 2.6.2

##### Variables Defined in Interface – N/A for Requirement 2.6.2

##### Types Defined in Interface – N/A for Requirement 2.6.2

##### GUI – PatientList.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| PatientList.pas | Removed code that checks for the existence of DoD data for the selected veteran. |

##### GUI Classes – N/A for Requirement 2.6.2

##### Current Form - N/A for Requirement 2.6.2

##### Modified Form – N/A for Requirement 2.6.2

##### Components on Form – N/A for Requirement 2.6.2

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| FMCvrListBox1PtClick | Listbox Item Click | Removed calls to check for DoD data if VistA patch 193 is installed. |
| btnCvrSelectPtClick | Button Click | Hide DoD tab when VistA patch 193 is installed. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| checkCCOWfordoddata | procedure | Removed call to check for DoD data when VistA patch 193 is installed. |
| Checkfordoddata | Procedure | Removed call to check for DoD data when VistA patch 193 is installed. |

##### Special References – N/A for Requirement 2.6.2

##### Class Events – N/A for Requirement 2.6.2

##### Class Methods – N/A for Requirement 2.6.2

##### Class Properties – N/A for Requirement 2.6.2

##### Uses Clause – N/A for Requirement 2.6.2

##### Forms – N/A for Requirement 2.6.2

##### Functions

Table 48: Functions

| Function Name | checkCCOWfordoddata | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Place code to hide the DoD info if Patch 193 is installed, show if any earlier patch is installed. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | btnCvrSelectPtClick | N/A |

| Function Name | checkCCOWfordoddata | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

**Function checkCCOWfordoddata: boolean;**

Var

x: integer;

Begin

result := false;

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := CCOWPatientIEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

result := false;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

result := true

End;

End;

End;

End;

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

**Function checkCCOWfordoddata: boolean;**

Var

x: integer;

Begin

result := false;

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := CCOWPatientIEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

result := false;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

result := true

End;

End;

End;

end; // IsPatchInstalled Patch 193 JRL 7/28/16

End;

Table 48: Functions

| Function Name | checkfordoddata | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Place code to hide the DoD info if Patch 193 is installed, show if any earlier patch is installed. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | TfrmPatientList.FMCvrListBox1PtClick | N/A |

| Function Name | checkfordoddata | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

**Procedure checkfordoddata;**

Var

x: integer;

Begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

// DOD data is available so do something here....

LabelStatus.Caption := LabelStatus.Caption + 'DoD data is available.' + char(13);

End;

End;

End;

End;

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

**Procedure checkfordoddata;**

Var

x: integer;

Begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin

If sitenum = '200' Then

Begin

// DOD data is available so do something here....

LabelStatus.Caption := LabelStatus.Caption + 'DoD data is available.' + char(13);

End;

end; // IsPatchInstalled Patch 193 JRL 7/28/16

End;

End;

End;

Table 48: Functions

| Function Name | btnCvrSelectPtClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Place code to hide the DoD tab if Patch 193 and disable the tab by default.. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | checkCCOWfordoddata | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

**Procedure TfrmPatientList.btnCvrSelectPtClick(Sender: TObject);**

Var

boIsDODData: Boolean;

buf: Array[0..7000] Of char;

foundflag: boolean;

msg: String;

PatientDFN: String;

x: integer;

xx: integer;

Function ptdeceasedcheck: boolean;

Var

diedtext: String;

Begin

diedtext := FMCvrGets1Pt.GetField('.351').FMDBExternal;

If diedtext <> '' Then

Begin

msg := 'This patient died ' + diedtext + char(13) + 'Do you wish to continue?';

StrPCopy(buf, msg);

If Application.MessageBox(buf, 'Deceased Patient', 4) = IDNo Then

Begin

patientien := '-1';

result := false;

exit;

End;

End;

result := true;

End;

Procedure ptcheckflags;

Var

x: integer;

Begin

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF TRIGGER POPUP';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF TRIGGER POPUP could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] = '1' Then

Begin

// Popup indicated

FormPtRecordFlags := TFormPtRecordFlags.create(frmPatientList);

FormPtRecordFlags.LabelPatientDFN.Caption := PatientDFN;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF HASFLG';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF HASFLG could not be accessed!');

frmMain.RPCBroker1.Results.Add('0'); // Let the user continue

End;

End;

If frmMain.RPCBroker1.Results.Count > 0 Then

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

FormPtRecordFlags.ORListBoxFlags.Items.Add(frmMain.RPCBroker1.Results[x]);

End;

FormPtRecordFlags.ORListBoxFlags.ItemIndex := 0;

FormPtRecordFlags.ORListBoxFlagsClick(Application);

End;

FormPtRecordFlags.ShowModal;

FormPtRecordFlags.Release;

End;

End;

Function checkCCOWfordoddata: boolean;

Var

x: integer;

Begin

result := false;

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := CCOWPatientIEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

result := false;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

result := true

End;

End;

End;

End;

Begin

If Patient\_ID.Text = '' Then

exit;

// Next line is sometimes needed when adding a new patient dialog has been executed

frmROFinder.visible := false;

frmMain.TabDoD.Enabled := True;

// Do the next section only for a double-click

// Wait for demographic data to load

btnCvrTimerPt.enabled := False;

PatientSensitive := 'N';

//First check if new patient selected in restricted list

//Requery just to be sure the one clicked is the one returned

If ORListBoxRestrictedPatients.ItemIndex <> -1 Then

ORListBoxRestrictedPatientsClick(Application);

If CCOWPatientIEN = '' Then

Begin

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD ACCESS';

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox3Pt.GetselectedRecord.IEN;

frmMain.RPCBroker1.Param[0].PType := literal;

PatientDFN := frmMain.RPCBroker1.Param[0].Value;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call

Except

On EBrokerError Do

Begin

enabled := True; //CodeCR124 - rpm 7/27/10

ShowMessageCAPRI('DG SENSITIVE RECORD ACCESS could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] <> '0' Then

Begin

If frmMain.RPCBroker1.Results[0] <> '3' Then

Begin

//2=regular patient

msg := '';

If frmMain.RPCBroker1.Results.Count > 0 Then

For xx := 1 To frmMain.RPCBroker1.Results.Count - 1 Do

msg := msg + frmMain.RPCBroker1.Results[xx] + char(13);

//xx:=0;

StrPCopy(buf, msg);

If ANURemoteProcedureCallInProgress = True Then

Repeat application.processmessages Until ANURemoteProcedureCallInProgress = False;

If Application.MessageBox(buf, 'Sensitive Record', 4) = IDNo Then

Begin

ProcessingTimer := False;

// Check that user should have access to this site in case they switched

// sites, don't normally have access and then cancelled a sensitive patient switch

If essoversion = true Then

Begin

foundflag := false;

If formESSOSelect.slUserSites.Count > 0 Then //CodeCR77

For x := 0 To formESSOSelect.slUserSites.Count - 1 Do

If uppercase(frmMain.piece(formESSOSelect.slUserSites[x], '^', 2)) = uppercase(frmMain.RPCBroker1.Server) Then

foundflag := true;

If foundflag <> true Then

Begin

ShowmessageCAPRI('You do not have direct access to this site. Please use FILE | SWITCH SITES and pick a different site prior to selecting a patient.');

btnCvrCancelPtClick(Application);

exit;

End;

End;

exit;

End;

// Send warning message to ISO

PatientSensitive := 'Y';

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD BULLETIN';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[2].Value := 'DVBA CAPRI GUI^Capri GUI (Broker)';

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[3].Value := ''; // Use the default for this parameter by leaving it blank

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

If frmMain.RPCBroker1.Results[0] <> '1' Then

Begin

ShowMessageCAPRI('There was a problem generating the DG SENSITIVE RECORD e-mail. Please notify IRM.');

End;

End;

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

If frmMain.RPCBroker1.Results[0] = '3' Then

Begin

msg := '';

If frmMain.RPCBroker1.Results.Count > 0 Then

For xx := 1 To frmMain.RPCBroker1.Results.Count - 1 Do

msg := msg + frmMain.RPCBroker1.Results[xx] + char(13);

ShowMessageCAPRI(msg);

processingtimer := false;

exit;

// Don't send warning message to ISO

End;

End;

End;

If CCOWPatientIEN <> '' Then

Begin

FMCvrListBox1Pt.ItemIndex := -1;

FMCvrListBox2Pt.ItemIndex := -1;

FMCvrListBox3Pt.ItemIndex := -1;

PatientIEN := CCOWPatientIEN;

PatientIENDoD := '';

boIsDODData:= CheckCCOWForDODData;

If boIsDODData Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

ccowclose.enabled := true;

exit;

End;

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox1Pt.Items[FMCvrListBox1Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox1Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

ModalResult := mrOK;

Close; //CodeCR124 - rpm 7/27/10

exit;

End;

End;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox2Pt.Items[FMCvrListBox2Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox2Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

TimerCloseit.enabled := true;

exit;

End;

End;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox3Pt.Items[FMCvrListBox3Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox3Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

ModalResult := mrOK;

Close; //CodeCR124 - rpm 7/27/10

exit;

End;

End;

End;

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

**Procedure TfrmPatientList.btnCvrSelectPtClick(Sender: TObject);**

Var

boIsDODData: Boolean;

buf: Array[0..7000] Of char;

foundflag: boolean;

msg: String;

PatientDFN: String;

x: integer;

xx: integer;

Function ptdeceasedcheck: boolean;

Var

diedtext: String;

Begin

diedtext := FMCvrGets1Pt.GetField('.351').FMDBExternal;

If diedtext <> '' Then

Begin

msg := 'This patient died ' + diedtext + char(13) + 'Do you wish to continue?';

StrPCopy(buf, msg);

If Application.MessageBox(buf, 'Deceased Patient', 4) = IDNo Then

Begin

patientien := '-1';

result := false;

exit;

End;

End;

result := true;

End;

Procedure ptcheckflags;

Var

x: integer;

Begin

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF TRIGGER POPUP';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF TRIGGER POPUP could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] = '1' Then

Begin

// Popup indicated

FormPtRecordFlags := TFormPtRecordFlags.create(frmPatientList);

FormPtRecordFlags.LabelPatientDFN.Caption := PatientDFN;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF HASFLG';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF HASFLG could not be accessed!');

frmMain.RPCBroker1.Results.Add('0'); // Let the user continue

End;

End;

If frmMain.RPCBroker1.Results.Count > 0 Then

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

FormPtRecordFlags.ORListBoxFlags.Items.Add(frmMain.RPCBroker1.Results[x]);

End;

FormPtRecordFlags.ORListBoxFlags.ItemIndex := 0;

FormPtRecordFlags.ORListBoxFlagsClick(Application);

End;

FormPtRecordFlags.ShowModal;

FormPtRecordFlags.Release;

End;

End;

Function checkCCOWfordoddata: boolean;

Var

x: integer;

Begin

result := false;

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := CCOWPatientIEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

result := false;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

result := true

End;

End;

End;

end; // IsPatchInstalled Patch 193 JRL 7/28/16

End;

Begin

If Patient\_ID.Text = '' Then

exit;

// Next line is sometimes needed when adding a new patient dialog has been executed

frmROFinder.visible := false;

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

frmMain.TabDoD.Enabled := False

else

frmMain.TabDoD.Enabled := True;

// Do the next section only for a double-click

// Wait for demographic data to load

btnCvrTimerPt.enabled := False;

PatientSensitive := 'N';

//First check if new patient selected in restricted list

//Requery just to be sure the one clicked is the one returned

If ORListBoxRestrictedPatients.ItemIndex <> -1 Then

ORListBoxRestrictedPatientsClick(Application);

If CCOWPatientIEN = '' Then

Begin

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD ACCESS';

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox3Pt.GetselectedRecord.IEN;

frmMain.RPCBroker1.Param[0].PType := literal;

PatientDFN := frmMain.RPCBroker1.Param[0].Value;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call

Except

On EBrokerError Do

Begin

enabled := True; //CodeCR124 - rpm 7/27/10

ShowMessageCAPRI('DG SENSITIVE RECORD ACCESS could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] <> '0' Then

Begin

If frmMain.RPCBroker1.Results[0] <> '3' Then

Begin

//2=regular patient

msg := '';

If frmMain.RPCBroker1.Results.Count > 0 Then

For xx := 1 To frmMain.RPCBroker1.Results.Count - 1 Do

msg := msg + frmMain.RPCBroker1.Results[xx] + char(13);

//xx:=0;

StrPCopy(buf, msg);

If ANURemoteProcedureCallInProgress = True Then

Repeat application.processmessages Until ANURemoteProcedureCallInProgress = False;

If Application.MessageBox(buf, 'Sensitive Record', 4) = IDNo Then

Begin

ProcessingTimer := False;

// Check that user should have access to this site in case they switched

// sites, don't normally have access and then cancelled a sensitive patient switch

If essoversion = true Then

Begin

foundflag := false;

If formESSOSelect.slUserSites.Count > 0 Then //CodeCR77

For x := 0 To formESSOSelect.slUserSites.Count - 1 Do

If uppercase(frmMain.piece(formESSOSelect.slUserSites[x], '^', 2)) = uppercase(frmMain.RPCBroker1.Server) Then

foundflag := true;

If foundflag <> true Then

Begin

ShowmessageCAPRI('You do not have direct access to this site. Please use FILE | SWITCH SITES and pick a different site prior to selecting a patient.');

btnCvrCancelPtClick(Application);

exit;

End;

End;

exit;

End;

// Send warning message to ISO

PatientSensitive := 'Y';

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD BULLETIN';

frmMain.RPCBroker1.Param[1].Value := PatientDFN;

frmMain.RPCBroker1.Param[1].PType := literal;

frmMain.RPCBroker1.Param[2].Value := 'DVBA CAPRI GUI^Capri GUI (Broker)';

frmMain.RPCBroker1.Param[2].PType := literal;

frmMain.RPCBroker1.Param[3].Value := ''; // Use the default for this parameter by leaving it blank

frmMain.RPCBroker1.Param[3].PType := literal;

frmMain.RPCBrokerCall;

frmMain.RPCBroker1.Call;

If frmMain.RPCBroker1.Results.Count > 0 Then

If frmMain.RPCBroker1.Results[0] <> '1' Then

Begin

ShowMessageCAPRI('There was a problem generating the DG SENSITIVE RECORD e-mail. Please notify IRM.');

End;

End;

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

If frmMain.RPCBroker1.Results[0] = '3' Then

Begin

msg := '';

If frmMain.RPCBroker1.Results.Count > 0 Then

For xx := 1 To frmMain.RPCBroker1.Results.Count - 1 Do

msg := msg + frmMain.RPCBroker1.Results[xx] + char(13);

ShowMessageCAPRI(msg);

processingtimer := false;

exit;

// Don't send warning message to ISO

End;

End;

End;

If CCOWPatientIEN <> '' Then

Begin

FMCvrListBox1Pt.ItemIndex := -1;

FMCvrListBox2Pt.ItemIndex := -1;

FMCvrListBox3Pt.ItemIndex := -1;

PatientIEN := CCOWPatientIEN;

PatientIENDoD := '';

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 193 and later

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

end

else

begin // Patch 192 and earlier

boIsDODData:= CheckCCOWForDODData;

If boIsDODData Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

end;

ccowclose.enabled := true;

exit;

End;

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox1Pt.Items[FMCvrListBox1Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox1Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 193 and later

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

end

else

begin // Patch 192 and earlier

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

end;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

ModalResult := mrOK;

Close; //CodeCR124 - rpm 7/27/10

exit;

End;

End;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox2Pt.Items[FMCvrListBox2Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox2Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 193 and later

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

end

else

begin // Patch 192 and earlier

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

end;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

TimerCloseit.enabled := true;

exit;

End;

End;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

Begin

If FMCvrListBox3Pt.Items[FMCvrListBox3Pt.ItemIndex] <> '' Then

Begin

If ptDeceasedCheck = false Then

Begin

processingtimer := false;

exit;

End;

ptcheckflags;

PatientIEN := FMCvrListBox3Pt.GetselectedRecord.IEN;

PatientIENDoD := '';

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 193 and later

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

end

else

begin // Patch 192 and earlier

If Pos('DOD DATA IS AVAILABLE.', Uppercase(LabelStatus.Caption)) > 0 Then

Begin

frmMain.TabDOD.Visible := True;

frmMain.TabDOD.TabVisible := True;

frmMain.Page95Control1.ActivePage := frmMain.TabDOD;

frmMain.Page95Control1.ActivePage := frmMain.TabCPExams;

End

Else

Begin

frmMain.TabDOD.Visible := False;

frmMain.TabDOD.TabVisible := False;

End;

end;

If CCOWMode = True Then

If frmMain.bitbtnCCOWLinkBroken.Visible = False Then

Begin

frmMain.BitBtnCCOWLinkBroken.Visible := False;

frmMain.BitBtnCCOWLink.Visible := False;

frmMain.BitBtnCCOWLinkChanging.Visible := True;

End;

ModalResult := mrOK;

Close; //CodeCR124 - rpm 7/27/10

exit;

End;

End;

End;

Table 48: Functions

| Function Name | FMCvrListBox1PtClick | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Removed calls to check for DoD data if VistA patch 193 is installed. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FMCvrListBox1PtClick | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

Procedure TfrmPatientList.FMCvrListBox1PtClick(Sender: TObject);

Procedure RemoveHDRFromList;

Var

x: integer;

Begin

// Remove HDR entry, if found

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) <> '-1' Then

For x := frmMain.RPCBroker1.Results.Count - 1 Downto 0 Do

If frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) = '200HD' Then

frmMain.RPCBroker1.Results.Delete(x);

// If the list is empty, make sure this is reflected.

If frmMain.RPCBroker1.Results.Count = 0 Then

frmMain.RPCBroker1.Results.Add('-1^No Sites Found');

End;

Procedure checkforflagsformessage;

Begin

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF TRIGGER POPUP';

frmMain.RPCBroker1.Param[1].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF TRIGGER POPUP could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] = '1' Then

Begin

LabelStatus.Caption := LabelStatus.Caption + 'Record flags found.' + char(13);

End;

End;

Procedure checkfordoddata;

Var

x: integer;

Begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

// DOD data is available so do something here....

LabelStatus.Caption := LabelStatus.Caption + 'DoD data is available.' + char(13);

End;

End;

End;

End;

Begin

If Uppercase(sender.classname) = 'TFMLISTBOX' Then

clickedyet := true;

sleep(250);

If ANURemoteProcedureCallInProgress = True Then

exit;

If (GroupBoxCvr1Pt.Visible = False) And (PanelCvr1Pt.Visible = False) Then

Begin

// Exit because the call is actually already in progress and the user

// kicked off another one

exit;

End;

GroupboxCvr1Pt.Visible := false;

btnCvrTimerPt.enabled := False;

btnCvrSelectPt.Enabled := false;

buttonOtherSites.Enabled := False;

LabelStatus.Caption := '';

PanelCvr1Pt.Visible := false;

ReFresh; //CodeCR124 - rpm 7/27/10

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD ACCESS';

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox3Pt.GetselectedRecord.IEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

enabled := True; //CodeCR124 - rpm 7/27/10

ShowMessageCAPRI('DG SENSITIVE RECORD ACCESS could not be accessed!');

End;

End;

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox3Pt.GetselectedRecord.IEN;

If frmMain.RPCBroker1.Results[0] = '0' Then

Begin

PanelCvr1Pt.Visible := False;

FMCvrNamePt.IENS := FMCvrGets1Pt.IENS;

FMCvrGets1Pt.GetAndFill;

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWCIRN FACLIST could not be accessed!');

End

End;

RemoveHDRFromList;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

ButtonOtherSites.Enabled := False;

End

Else

Begin

ButtonOtherSites.Enabled := True;

End;

CheckForDODData;

checkforflagsformessage;

GroupBoxCvr1Pt.Visible := True; // Hide info on sensitive pt.'s

If FMCvrGets1Pt.GetField('.351').FMDBExternal <> '' Then

Begin

LabelStatus.Caption := LabelStatus.Caption + 'Patient is deceased.' + char(13);

LabelStatus.Visible := True;

End;

End

Else

Begin

GroupBoxCvr1Pt.Visible := False; // Hide info on sensitive pt.'s

FMCvrNamePt.IENS := FMCvrGets1Pt.IENS;

FMCvrGets1Pt.GetAndFill;

ButtonOtherSites.Enabled := True;

CheckForDODData;

checkforflagsformessage;

PanelCvr1Pt.Visible := True;

End;

LabelCvrCNumberPt.Visible := True;

LabelCvrSSN.Visible := True;

LabelCvrICNPt.Visible := True;

LabelCvrWard.Visible := True;

FMCvrAgePt.Caption := FMCvrAgePt.Caption + ' y/o';

// Check for errors after the server call.

If FMCvrGets1Pt.ErrorList.Count > 0 Then

FMCvrGets1Pt.DisplayErrors;

If FMCvrListBox1Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If FMCvrListBox2Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If FMCvrListBox3Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If btnCvrSelectPt.enabled = true Then

btnCvrDemographicsPt.Enabled := true;

enabled := True; //CodeCR124 - rpm 7/27/10

If ReadOnlyMode = False Then

Begin

If Pos('@', userfilemancode) > 0 Then

buttonEdit.enabled := true;

If Pos('u', userfilemancode) > 0 Then

buttonEdit.enabled := true;

If Pos('U', userfilemancode) > 0 Then

buttonEdit.enabled := true;

End;

End;

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

Procedure TfrmPatientList.FMCvrListBox1PtClick(Sender: TObject);

Procedure RemoveHDRFromList;

Var

x: integer;

Begin

// Remove HDR entry, if found

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) <> '-1' Then

For x := frmMain.RPCBroker1.Results.Count - 1 Downto 0 Do

If frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) = '200HD' Then

frmMain.RPCBroker1.Results.Delete(x);

// If the list is empty, make sure this is reflected.

If frmMain.RPCBroker1.Results.Count = 0 Then

frmMain.RPCBroker1.Results.Add('-1^No Sites Found');

End;

Procedure checkforflagsformessage;

Begin

exit;

frmMain.RPCBroker1.RemoteProcedure := 'ORPRF TRIGGER POPUP';

frmMain.RPCBroker1.Param[1].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[1].PType := literal;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

ShowMessageCAPRI('ORPRF TRIGGER POPUP could not be accessed!');

End;

End;

If frmMain.RPCBroker1.Results[0] = '1' Then

Begin

LabelStatus.Caption := LabelStatus.Caption + 'Record flags found.' + char(13);

End;

End;

Procedure checkfordoddata;

Var

x: integer;

Begin

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

exit;

End

End;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

// Do nothing

End

Else

Begin

For x := 0 To frmMain.RPCBroker1.Results.Count - 1 Do

Begin

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1) + ' ', 1, 8);

Copy(frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 2) + ' ', 1, 25);

sitenum := frmMain.Piece(frmMain.RPCBroker1.Results[x], '^', 1);

If sitenum = '200' Then

Begin

// DOD data is available so do something here....

LabelStatus.Caption := LabelStatus.Caption + 'DoD data is available.' + char(13);

End;

End;

End;

End;

Begin

If Uppercase(sender.classname) = 'TFMLISTBOX' Then

clickedyet := true;

sleep(250);

If ANURemoteProcedureCallInProgress = True Then

exit;

If (GroupBoxCvr1Pt.Visible = False) And (PanelCvr1Pt.Visible = False) Then

Begin

// Exit because the call is actually already in progress and the user

// kicked off another one

exit;

End;

GroupboxCvr1Pt.Visible := false;

btnCvrTimerPt.enabled := False;

btnCvrSelectPt.Enabled := false;

buttonOtherSites.Enabled := False;

LabelStatus.Caption := '';

PanelCvr1Pt.Visible := false;

ReFresh; //CodeCR124 - rpm 7/27/10

frmMain.RPCBroker1.RemoteProcedure := 'DG SENSITIVE RECORD ACCESS';

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

frmMain.RPCBroker1.Param[0].Value := FMCvrListBox3Pt.GetselectedRecord.IEN;

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

enabled := True; //CodeCR124 - rpm 7/27/10

ShowMessageCAPRI('DG SENSITIVE RECORD ACCESS could not be accessed!');

End;

End;

If FMCvrListBox1Pt.Visible = True Then

If FMCvrListBox1Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox1Pt.GetselectedRecord.IEN;

If FMCvrListBox2Pt.Visible = True Then

If FMCvrListBox2Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox2Pt.GetselectedRecord.IEN;

If FMCvrListBox3Pt.Visible = True Then

If FMCvrListBox3Pt.ItemIndex > -1 Then

FMCvrGets1Pt.IENS := FMCvrListBox3Pt.GetselectedRecord.IEN;

If frmMain.RPCBroker1.Results[0] = '0' Then

Begin

PanelCvr1Pt.Visible := False;

FMCvrNamePt.IENS := FMCvrGets1Pt.IENS;

FMCvrGets1Pt.GetAndFill;

frmMain.RPCBroker1.RemoteProcedure := 'ORWCIRN FACLIST';

frmMain.RPCBroker1.Param[0].Value := frmMain.Piece(FMCvrGets1Pt.IENS, ',', 1);

frmMain.RPCBroker1.Param[0].PType := literal;

frmMain.RPCBrokerCall;

Try

frmMain.RPCBroker1.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

frmMain.AnimateLogo(False);

frmMain.StatusBarLoadPt.Caption := 'RPC ORWCIRN FACLIST could not be accessed!';

frmMain.StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('ORWCIRN FACLIST could not be accessed!');

End

End;

RemoveHDRFromList;

If frmMain.Piece(frmMain.RPCBroker1.Results[0], '^', 1) = '-1' Then

Begin

// Only local data or other problem

ButtonOtherSites.Enabled := False;

End

Else

Begin

ButtonOtherSites.Enabled := True;

End;

If not IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 JRL 8/15/16

CheckForDODData;

checkforflagsformessage;

GroupBoxCvr1Pt.Visible := True; // Hide info on sensitive pt.'s

If FMCvrGets1Pt.GetField('.351').FMDBExternal <> '' Then

Begin

LabelStatus.Caption := LabelStatus.Caption + 'Patient is deceased.' + char(13);

LabelStatus.Visible := True;

End;

End

Else

Begin

GroupBoxCvr1Pt.Visible := False; // Hide info on sensitive pt.'s

FMCvrNamePt.IENS := FMCvrGets1Pt.IENS;

FMCvrGets1Pt.GetAndFill;

ButtonOtherSites.Enabled := True;

If not IsPatchInstalled('DVBA\*2.7\*193') then // Patch 193 JRL 8/15/16

CheckForDODData;

checkforflagsformessage;

PanelCvr1Pt.Visible := True;

End;

LabelCvrCNumberPt.Visible := True;

LabelCvrSSN.Visible := True;

LabelCvrICNPt.Visible := True;

LabelCvrWard.Visible := True;

FMCvrAgePt.Caption := FMCvrAgePt.Caption + ' y/o';

// Check for errors after the server call.

If FMCvrGets1Pt.ErrorList.Count > 0 Then

FMCvrGets1Pt.DisplayErrors;

If FMCvrListBox1Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If FMCvrListBox2Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If FMCvrListBox3Pt.ItemIndex > -1 Then

btnCvrSelectPt.Enabled := True;

If btnCvrSelectPt.enabled = true Then

btnCvrDemographicsPt.Enabled := true;

enabled := True; //CodeCR124 - rpm 7/27/10

If ReadOnlyMode = False Then

Begin

If Pos('@', userfilemancode) > 0 Then

buttonEdit.enabled := true;

If Pos('u', userfilemancode) > 0 Then

buttonEdit.enabled := true;

If Pos('U', userfilemancode) > 0 Then

buttonEdit.enabled := true;

End;

End;

##### Dialog – N/A for Requirement 2.6.2

##### GUI – Main.pas

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| Main.pas | Remove code that displays and updates the DoD tab if VistA Patch 193 is installed. Because of deployment issues, code and GUI for the DoD tab was not removed. It was deprecated instead. Only sites with VistA Patch 192 and earlier will see the DoD tab. Patch 193 and beyond will not see this tab. |

##### GUI Classes – N/A for Requirement 2.6.2

##### Current Form – N/A for Requirement 2.6.2

##### Modified Form – N/A for Requirement 2.6.2

##### Components on Form – N/A for Requirement 2.6.2

##### Events

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| Page95Control1Change | Tab Change | Don’t allow user to change to DoD tab |
| actFileTransmitVirtualVAExecute | Menu Option Selection | Remove DoD specific information. |

##### Methods

Table 41: Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| GetFooterDescription | Function | Format Report Footer with DoD information – add patch specific code. |
| FormatReportBody | Procedure | Format Report Footer with DoD information – add patch specific code. |

##### Special References – N/A for Requirement 2.6.2

##### Class Events – N/A for Requirement 2.6.2

##### Class Methods – N/A for Requirement 2.6.2

##### Class Properties – N/A for Requirement 2.6.2

##### Uses Clause – N/A for Requirement 2.6.2

##### Forms – N/A for Requirement 2.6.2

##### Functions

Table 48: Functions

| Function Name | Page95Control1Change | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Remove DoD information if Patch 193 is installed. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

**Procedure TfrmMain.Page95Control1Change(Sender: TObject);**

Var

x: integer;

StationNo: String; //required for Local FHIE connections

Begin

ReverseExamRequestSortOrder := False; // CodeCR708 rpk 8/27/2015

// ReverseAdminSortOrder := False; // CodeCR708 rpk 8/27/2015

// if Page95Control1.ActivePage = AdminDocuments then // CodeCR708 rpk 8/27/2015

// btnSortAdmin.Caption := 'Sort Oldest on top ^'; // CodeCR708 rpk 8/27/2015

StationNo := '';

TabDOD.Enabled := True; // DoD tab is sometimes greying out. This forces it to normal.

SetFont; // Force update of font just in case

frmMain.FormResize(Application);

If Page95Control1.ActivePage = TabCPExams Then

Begin

cbxSortExamReq.ItemIndex := 0; // CodeCR708 rpk 8/27/2015

ExamRequestRefreshClick(Application);

End;

If Page95Control1.ActivePage = Tab7131Request Then

Begin

If FMSeventyOne31RequestListbox.ItemIndex = -1 Then

Begin

btnGenerate7131report.Enabled := False;

btnView7131.Enabled := False;

End

Else

Begin

btnGenerate7131report.Enabled := True;

btnView7131.Enabled := True;

End;

End;

If Page95Control1.ActivePage = TabCPWorksheets Then

Begin

ButtonIPRRefreshClick(Application);

End;

If Page95Control1.ActivePage = TabHealthSummaries Then

Begin

If ESSOVersion = True Then

buttonRDV.Enabled := False;

End;

If Page95Control1.ActivePage = TabClinicalDocuments Then

Begin

// -MER CodeCR177 3/2011 restore the tabs with hotkeys

Tab95Control1.Tabs.Clear;

Tab95Control1.Tabs.Add('&1) Notes');

Tab95Control1.Tabs.Add('&2) Discharge Summaries');

Tab95Control1.Tabs.Add('&3) Consults');

Tab95Control1.Tabs.Add('&4) Vitals');

Tab95Control1.Tabs.Add('&5) Meds');

Tab95Control1.Tabs.Add('&6) Labs');

Tab95Control1.Tabs.Add('&7) Imaging');

Tab95Control1.Tabs.Add('&8) Diet');

Tab95Control1.Tabs.Add('&9) Nutritional Assessment');

Tab95Control1.Tabs.Add('&U) Order Summary');

Tab95Control1.Tabs.Add('&Y) Procedures');

Tab95Control1.Tabs.Add('&Z) Problem List');

If Tab95Control1.TabIndex < 0 Then

Tab95Control1.TabIndex := 0;

Tab95Control1Change(Application);

If ScreenReaderActive = True Then

Page95Control1.SetFocus;

end else begin

// -MER CodeCR177 3/2011 Clear the tabs with hotkeys so they won't trap keystrokes

Tab95Control1.Tabs.Clear;

end;

If Page95Control1.ActivePage <> TabClinicalDocuments Then

Begin

If FormReportBuilder <> Nil Then

FormReportBuilder.Hide;

End;

If Page95Control1.ActivePage = TabVistAWeb Then

Begin

If VistAWebLoaded <> True Then

Begin

btnVistAWebHomeClick(Application);

VistAWebLoaded := True;

End;

End;

If Page95Control1.ActivePage = TabDOD Then

Begin

PanelDODGrids.Visible := False;

If TabDOD.enabled = False Then

exit;

If LabelDoDInfo.Caption = '\*\* NOT CONNECTED \*\*' Then

Begin

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating DoD tab.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

//Set DOD Broker division and get BSE token from home server

RPCBrokerDOD.ANUstrDivisionHome := RPCBroker1.ANUstrDivisionHome;

RPCBrokerDOD.Connect2HomeServer(RPCBroker1.ANUAccessCodeHome,

RPCBroker1.ANUVerifyCodeHome,

RPCBroker1.ANUstrServerHome,

RPCBroker1.ANUstrPortHome,

False);

if NOT RPCBrokerDOD.CreateContext('DVBA CAPRI GUI') then

begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

end;

RPCBrokerDOD.ANUFHIEConnection := True;

//determine station number required for local FHIE connections;

// short-circuit when no station number

if NOT ESSOVersion then

begin

StationNo := Piece(GetDivision(ESSOVersion, RPCBrokerDOD), '^', 3);

if StationNo = '' then

begin

RPCBrokerDOD.ANUFHIEConnection := False;

RPCBrokerDOD.Connected := False;

ShowMessageCapri('Could not determine local station number for FHIE connection.');

Exit;

end;

end;

RPCBrokerDOD.Connected := False;

sleep(1000);

RPCBrokerDOD.RPCTimeLimit := 28800;

RPCBrokerDOD.Server := Piece(FHIESiteLocation, ',', 1);

RPCBrokerDOD.ListenerPort := StrtoInt(Piece(FHIESiteLocation, ',', 2));

RPCBrokerDOD.ANUUserSpecs :=

AuthorSSN + '^' +

AuthorName + '^' +

UserDivision + '^' + // Site

UserDivisionNumber + '^' + // Site num

AuthorIEN;

//connect to remote server with BSE token

if NOT RPCBrokerDOD.Connect2RemoteServer(RPCBrokerDOD.Server,

IntToStr(RPCBrokerDOD.ListenerPort),

False, True, StationNo) then

begin

AnimateLogo(False);

RPCBrokerDOD.Connected := False;

StatusBarLoadPt.Caption := 'FHIE Server unavailable.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

RPCBrokerDOD.Connected := False;

ShowMessageCAPRI('FHIE Server cannot be accessed!');

//RPCBroker1.Connected := true; BSE mod - replace with connecttoserver

if NOT RPCBroker1.Connected then

ConnectToServer('DVBA CAPRI GUI');

RPCBrokerDOD.Connected := False;

ESSOConnecting := True;

exit;

end;

If Not RPCBrokerDOD.CreateContext('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

//LoadTypesFromServer

RPCBrokerDOD.RemoteProcedure := 'DVBAB DOD REPORT TYPES';

RPCBrokerDODCall;

Try

RPCBrokerDOD.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DOD REPORT TYPES could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DOD REPORT TYPES could not be accessed!');

End;

End;

ORListBoxDODReportTypes.Items := RPCBrokerDOD.Results;

StatusBarLoadPt.Caption := 'Getting info message.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

RPCBrokerDOD.RemoteProcedure := 'DVBAB DOD INFO';

RPCBrokerDODCall;

Try

RPCBrokerDOD.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DOD INFO could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DOD INFO could not be accessed!');

RPCBrokerDOD.Connected := False;

RPCBrokerDOD.ANUFHIEConnection := False;

RPCBrokerDOD.Connected := False;

ESSOConnecting := True;

exit;

End;

End;

LabelDoDInfo.Caption := RPCBrokerDOD.Results[0];

screen.cursor := crHourglass;

RPCBrokerDOD.Connected := False;

sleep(1000);

ESSOConnecting := True;

For x := 1 To 500 Do

application.processmessages;

// The following code was changed because Togus was locking after an FHIE call

RPCBroker1.Connected := False;

sleep(1000);

//RPCBroker1.Connected := true; BSE mod - rpm 3/16/09

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('DOD1: Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('DOD2: Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

Screen.cursor := crDefault;

ESSOConnecting := False;

//////////////////////////////////////////////////////////////////////////////

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

RichEditDODReport.Lines.Clear;

PanelDODGrids.Visible := False;

ORListBoxDODReportTypes.ItemIndex := -1;

AnimateLogo(False);

RPCBrokerDOD.ANUFHIEConnection := False;

End;

RPCBrokerDOD.ANUFHIEConnection := False;

If Page95Control1.ActivePage = TabVocRehab Then //CodeCR347 - jrl 5/22/12

Begin //CodeCR347 - jrl 5/22/12

// create VocRehab class if not done already //CodeCR347 - jrl 5/22/12

if not Assigned(VocRehab) then //CodeCR347 - jrl 5/22/12

VocRehab := TVocRehab.Create; //CodeCR347 - jrl 5/22/12

// Setup the Medical Services Request list for display on VocRehab tab //CodeCR347 - jrl 5/22/12

VocRehab.SetupMedicalServicesList; //CodeCR347 - jrl 5/22/12

if VAUtils.ScreenReaderActive then

begin

edtHeader.TabStop := TRUE;

edtColumnHeader.TabStop := TRUE;

edtHeader.SetFocus;

end

else

begin

edtHeader.TabStop := FALSE;

edtColumnHeader.TabStop := FALSE;

lbVocRehab.SetFocus; //CodeCR347 - jrl 5/22/12

end;

if lbVocRehab.ItemIndex <> -1 then //CodeCR347 - jrl 5/22/12

btnVREditRequest.Enabled := TRUE; //CodeCR347 - jrl 5/22/12

End; //CodeCR347 - jrl 5/22/12

End;

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

**Procedure TfrmMain.Page95Control1Change(Sender: TObject);**

Var

x: integer;

StationNo: String; //required for Local FHIE connections

Begin

ReverseExamRequestSortOrder := False; // CodeCR708 rpk 8/27/2015

// ReverseAdminSortOrder := False; // CodeCR708 rpk 8/27/2015

// if Page95Control1.ActivePage = AdminDocuments then // CodeCR708 rpk 8/27/2015

// btnSortAdmin.Caption := 'Sort Oldest on top ^'; // CodeCR708 rpk 8/27/2015

StationNo := '';

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If IsPatchInstalled('DVBA\*2.7\*193') then

TabDOD.Enabled := FALSE; // Patch 193 and later

else

TabDOD.Enabled := True; // Patch 192 and earlier // DoD tab is sometimes greying out. This forces it to normal.

SetFont; // Force update of font just in case

frmMain.FormResize(Application);

If Page95Control1.ActivePage = TabCPExams Then

Begin

cbxSortExamReq.ItemIndex := 0; // CodeCR708 rpk 8/27/2015

ExamRequestRefreshClick(Application);

End;

If Page95Control1.ActivePage = Tab7131Request Then

Begin

If FMSeventyOne31RequestListbox.ItemIndex = -1 Then

Begin

btnGenerate7131report.Enabled := False;

btnView7131.Enabled := False;

End

Else

Begin

btnGenerate7131report.Enabled := True;

btnView7131.Enabled := True;

End;

End;

If Page95Control1.ActivePage = TabCPWorksheets Then

Begin

ButtonIPRRefreshClick(Application);

End;

If Page95Control1.ActivePage = TabHealthSummaries Then

Begin

If ESSOVersion = True Then

buttonRDV.Enabled := False;

End;

If Page95Control1.ActivePage = TabClinicalDocuments Then

Begin

// -MER CodeCR177 3/2011 restore the tabs with hotkeys

Tab95Control1.Tabs.Clear;

Tab95Control1.Tabs.Add('&1) Notes');

Tab95Control1.Tabs.Add('&2) Discharge Summaries');

Tab95Control1.Tabs.Add('&3) Consults');

Tab95Control1.Tabs.Add('&4) Vitals');

Tab95Control1.Tabs.Add('&5) Meds');

Tab95Control1.Tabs.Add('&6) Labs');

Tab95Control1.Tabs.Add('&7) Imaging');

Tab95Control1.Tabs.Add('&8) Diet');

Tab95Control1.Tabs.Add('&9) Nutritional Assessment');

Tab95Control1.Tabs.Add('&U) Order Summary');

Tab95Control1.Tabs.Add('&Y) Procedures');

Tab95Control1.Tabs.Add('&Z) Problem List');

If Tab95Control1.TabIndex < 0 Then

Tab95Control1.TabIndex := 0;

Tab95Control1Change(Application);

If ScreenReaderActive = True Then

Page95Control1.SetFocus;

end else begin

// -MER CodeCR177 3/2011 Clear the tabs with hotkeys so they won't trap keystrokes

Tab95Control1.Tabs.Clear;

end;

If Page95Control1.ActivePage <> TabClinicalDocuments Then

Begin

If FormReportBuilder <> Nil Then

FormReportBuilder.Hide;

End;

If Page95Control1.ActivePage = TabVistAWeb Then

Begin

If VistAWebLoaded <> True Then

Begin

btnVistAWebHomeClick(Application);

VistAWebLoaded := True;

End;

End;

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 192 and earlier

If Page95Control1.ActivePage = TabDOD Then

Begin

PanelDODGrids.Visible := False;

If TabDOD.enabled = False Then

exit;

If LabelDoDInfo.Caption = '\*\* NOT CONNECTED \*\*' Then

Begin

AnimateLogo(True);

StatusBarLoadPt.Caption := 'Populating DoD tab.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

//Set DOD Broker division and get BSE token from home server

RPCBrokerDOD.ANUstrDivisionHome := RPCBroker1.ANUstrDivisionHome;

RPCBrokerDOD.Connect2HomeServer(RPCBroker1.ANUAccessCodeHome,

RPCBroker1.ANUVerifyCodeHome,

RPCBroker1.ANUstrServerHome,

RPCBroker1.ANUstrPortHome,

False);

if NOT RPCBrokerDOD.CreateContext('DVBA CAPRI GUI') then

begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

end;

RPCBrokerDOD.ANUFHIEConnection := True;

//determine station number required for local FHIE connections;

// short-circuit when no station number

if NOT ESSOVersion then

begin

StationNo := Piece(GetDivision(ESSOVersion, RPCBrokerDOD), '^', 3);

if StationNo = '' then

begin

RPCBrokerDOD.ANUFHIEConnection := False;

RPCBrokerDOD.Connected := False;

ShowMessageCapri('Could not determine local station number for FHIE connection.');

Exit;

end;

end;

RPCBrokerDOD.Connected := False;

sleep(1000);

RPCBrokerDOD.RPCTimeLimit := 28800;

RPCBrokerDOD.Server := Piece(FHIESiteLocation, ',', 1);

RPCBrokerDOD.ListenerPort := StrtoInt(Piece(FHIESiteLocation, ',', 2));

RPCBrokerDOD.ANUUserSpecs :=

AuthorSSN + '^' +

AuthorName + '^' +

UserDivision + '^' + // Site

UserDivisionNumber + '^' + // Site num

AuthorIEN;

//connect to remote server with BSE token

if NOT RPCBrokerDOD.Connect2RemoteServer(RPCBrokerDOD.Server,

IntToStr(RPCBrokerDOD.ListenerPort),

False, True, StationNo) then

begin

AnimateLogo(False);

RPCBrokerDOD.Connected := False;

StatusBarLoadPt.Caption := 'FHIE Server unavailable.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

RPCBrokerDOD.Connected := False;

ShowMessageCAPRI('FHIE Server cannot be accessed!');

//RPCBroker1.Connected := true; BSE mod - replace with connecttoserver

if NOT RPCBroker1.Connected then

ConnectToServer('DVBA CAPRI GUI');

RPCBrokerDOD.Connected := False;

ESSOConnecting := True;

exit;

end;

If Not RPCBrokerDOD.CreateContext('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

//LoadTypesFromServer

RPCBrokerDOD.RemoteProcedure := 'DVBAB DOD REPORT TYPES';

RPCBrokerDODCall;

Try

RPCBrokerDOD.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DOD REPORT TYPES could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DOD REPORT TYPES could not be accessed!');

End;

End;

ORListBoxDODReportTypes.Items := RPCBrokerDOD.Results;

StatusBarLoadPt.Caption := 'Getting info message.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

RPCBrokerDOD.RemoteProcedure := 'DVBAB DOD INFO';

RPCBrokerDODCall;

Try

RPCBrokerDOD.Call;

Except

On EBrokerError Do

Begin

ANURemoteProcedureCallInProgress := False;

AnimateLogo(False);

StatusBarLoadPt.Caption := 'RPC DVBAB DOD INFO could not be accessed!';

StatusBarLoadPt.Repaint;

Application.Processmessages;

ShowMessageCAPRI('DVBAB DOD INFO could not be accessed!');

RPCBrokerDOD.Connected := False;

RPCBrokerDOD.ANUFHIEConnection := False;

RPCBrokerDOD.Connected := False;

ESSOConnecting := True;

exit;

End;

End;

LabelDoDInfo.Caption := RPCBrokerDOD.Results[0];

screen.cursor := crHourglass;

RPCBrokerDOD.Connected := False;

sleep(1000);

ESSOConnecting := True;

For x := 1 To 500 Do

application.processmessages;

// The following code was changed because Togus was locking after an FHIE call

RPCBroker1.Connected := False;

sleep(1000);

//RPCBroker1.Connected := true; BSE mod - rpm 3/16/09

If Not ConnectToServer('DVBA CAPRI GUI') Then

Begin

ShowMessageCAPRI('DOD1: Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

If RPCBroker1.Connected = False Then

Begin

ShowMessageCAPRI('DOD2: Could not use option "DVBA CAPRI GUI!"');

application.terminate;

End;

Screen.cursor := crDefault;

ESSOConnecting := False;

//////////////////////////////////////////////////////////////////////////////

StatusBarLoadPt.Caption := 'Ready.';

StatusBarLoadPt.Repaint;

Application.Processmessages;

End;

RichEditDODReport.Lines.Clear;

PanelDODGrids.Visible := False;

ORListBoxDODReportTypes.ItemIndex := -1;

AnimateLogo(False);

RPCBrokerDOD.ANUFHIEConnection := False;

End;

end;

RPCBrokerDOD.ANUFHIEConnection := False;

If Page95Control1.ActivePage = TabVocRehab Then //CodeCR347 - jrl 5/22/12

Begin //CodeCR347 - jrl 5/22/12

// create VocRehab class if not done already //CodeCR347 - jrl 5/22/12

if not Assigned(VocRehab) then //CodeCR347 - jrl 5/22/12

VocRehab := TVocRehab.Create; //CodeCR347 - jrl 5/22/12

// Setup the Medical Services Request list for display on VocRehab tab //CodeCR347 - jrl 5/22/12

VocRehab.SetupMedicalServicesList; //CodeCR347 - jrl 5/22/12

if VAUtils.ScreenReaderActive then

begin

edtHeader.TabStop := TRUE;

edtColumnHeader.TabStop := TRUE;

edtHeader.SetFocus;

end

else

begin

edtHeader.TabStop := FALSE;

edtColumnHeader.TabStop := FALSE;

lbVocRehab.SetFocus; //CodeCR347 - jrl 5/22/12

end;

if lbVocRehab.ItemIndex <> -1 then //CodeCR347 - jrl 5/22/12

btnVREditRequest.Enabled := TRUE; //CodeCR347 - jrl 5/22/12

End; //CodeCR347 - jrl 5/22/12

End;

Table 48: Functions

| Function Name | GetFooterDescription | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Remove DoD related report information for Patch 193 and later. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | GetFooterDescription | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

{=========================================================================

GetFooterDescription

This function evaluates the selected tab or form and returns footer

description text for reporting. This code refactored from

actFilePrintExecute and contains a number of short-circuits.

Input: Sender - form object

Output: footDescrip - footer description passed by reference

ErrorText - error message text

function result returns True on success; otherwise, returns False

CodeCR353 - rpm 4/30/12

========================================================================}

**function TfrmMain.GetFooterDescription(Sender: TObject; var footDescrip: String; var ErrorText: String): Boolean;**

begin

if Assigned(Sender) and (Sender = PNCSForm) then

begin {PNCS Exam report}

Page95Control1.ActivePage := TabCPWorksheets;

footDescrip := 'UNSIGNED/DRAFT C&P Template';

end {PNCS Exam report}

else if Assigned(Sender) and ((Sender = FormTIUDisplay) OR (Sender = TIUSignForm)) Then // CodeCR425 JRL 11/13/12 -- added TIUSignForm option

begin {CPRS Document view}

Page95Control1.ActivePage := TabCPWorksheets;

footdescrip := 'CLIN DOC: Progress Note';

end {CPRS Document view}

else If Assigned(Sender) and (Sender = frmExamDetails) Then

Begin {Exam Details form}

If frmExamDetails.RichEditExamReport.Visible = False Then

ErrorText := 'This exam cannot be printed yet.'

else

footdescrip := 'C&P Exam Results: ' + frmExamDetails.FMExamType.Text;

End {Exam Details form}

else If Assigned(Sender) and (Sender = frmGenerateBlankTemplate) Then

Begin {Blank Template form}

footdescrip := 'Blank Template: ' + Piece(frmGenerateBlankTemplate.ListBoxTemplates.Items[frmGenerateBlankTemplate.ListBoxTemplates.ItemIndex], '^', 1);

End {Blank Template form}

else

begin {All other senders}

if Page95Control1.ActivePage = TabReports Then

begin

if ORReportsAvailable.ItemIndex = -1 Then

begin

ErrorText := 'Please select a report type first.';

end

else

footdescrip := 'REPORT: ' + ORReportsAvailable.Items[OrReportsAvailable.ItemIndex];

end

else if Page95Control1.ActivePage = TabDoD then

begin

if ORListBoxDoDReportTypes.ItemIndex = -1 then

begin

ErrorText := 'Please select a report type first.';

end

else

footdescrip := 'FHIE: ' + Piece(ORListBoxDoDReportTypes.Items[ORListBoxDoDReportTypes.ItemIndex], '^', 1);

end

else if (Page95Control1.ActivePage = AdminDocuments) then

begin

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&V) Address') then

footdescrip := 'Address';

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&W) Appointments') Then

footdescrip := lblApptStatus.Caption;

end

else if Page95Control1.ActivePage = TabClinicalDocuments Then

begin {Clinical Documents}

if ((lstDocs.ItemIndex = -1) And (DocType <> 'DIET')) And (DocType <> 'MULTI') Then

begin

ErrorText := 'Please select a clinical document to print.';

end

else

begin {Clinical Document Types}

If DocType = 'PN' Then

footdescrip := 'CLIN DOC: Progress Note'

else If DocType = 'DS' Then

footdescrip := 'CLIN DOC: Discharge Summary'

else If DocType = 'CN' Then

footdescrip := 'CLIN DOC: Consult'

else If DocType = 'VS' Then

footdescrip := 'CLIN DOC: Vital Signs (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'MEDS' Then

footdescrip := 'CLIN DOC: Medications (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'LAB' Then

footdescrip := 'CLIN DOC: Labs (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'IMAGING' Then

footdescrip := 'CLIN DOC: Imaging'

else If DocType = 'DIET' Then

footdescrip := 'CLIN DOC: Diet'

else If DocType = 'NUTASSESS' Then

footdescrip := 'CLIN DOC: Nutritional Assessment (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'ORDERS' Then

footdescrip := 'CLIN DOC: Orders (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'PROCEDURES' Then

footdescrip := 'CLIN DOC: Procedure'

else If DocType = 'PROBLEM LIST' Then

footdescrip := 'CLIN DOC: Problem List (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'MULTI' Then

footdescrip := 'CLIN DOC: Multi-Document';

end; {Clinical Document Types}

End {Clinical Documents}

else If Page95Control1.ActivePage = TabHealthSummaries Then

Begin {Health Summaries}

if (ORHealthSummaryUserList.ItemIndex = -1) and

(HSMemoLocal.Lines.Count = 0) Then

begin

ErrorText := 'Please select a health summary type first.';

end

else

begin

If (ORHealthSummaryUserList.ItemIndex > -1) Then

footdescrip := 'HEALTH SUMMARY (' + Tab95Control2.Tabs[Tab95Control2.TabIndex] + '): ' + ORHealthSummaryUserList.Items[OrHealthSummaryUserList.ItemIndex]

Else

footdescrip := 'HEALTH SUMMARY (' + Tab95Control2.Tabs[Tab95Control2.TabIndex] + '): Adhoc Report';

end;

End; {Health Summaries}

End; {All other senders}

if (ErrorText <> '') then

Result := False

else

Result := True;

end;

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

{=========================================================================

GetFooterDescription

This function evaluates the selected tab or form and returns footer

description text for reporting. This code refactored from

actFilePrintExecute and contains a number of short-circuits.

Input: Sender - form object

Output: footDescrip - footer description passed by reference

ErrorText - error message text

function result returns True on success; otherwise, returns False

CodeCR353 - rpm 4/30/12

========================================================================}

**function TfrmMain.GetFooterDescription(Sender: TObject; var footDescrip: String; var ErrorText: String): Boolean;**

begin

if Assigned(Sender) and (Sender = PNCSForm) then

begin {PNCS Exam report}

Page95Control1.ActivePage := TabCPWorksheets;

footDescrip := 'UNSIGNED/DRAFT C&P Template';

end {PNCS Exam report}

else if Assigned(Sender) and ((Sender = FormTIUDisplay) OR (Sender = TIUSignForm)) Then // CodeCR425 JRL 11/13/12 -- added TIUSignForm option

begin {CPRS Document view}

Page95Control1.ActivePage := TabCPWorksheets;

footdescrip := 'CLIN DOC: Progress Note';

end {CPRS Document view}

else If Assigned(Sender) and (Sender = frmExamDetails) Then

Begin {Exam Details form}

If frmExamDetails.RichEditExamReport.Visible = False Then

ErrorText := 'This exam cannot be printed yet.'

else

footdescrip := 'C&P Exam Results: ' + frmExamDetails.FMExamType.Text;

End {Exam Details form}

else If Assigned(Sender) and (Sender = frmGenerateBlankTemplate) Then

Begin {Blank Template form}

footdescrip := 'Blank Template: ' + Piece(frmGenerateBlankTemplate.ListBoxTemplates.Items[frmGenerateBlankTemplate.ListBoxTemplates.ItemIndex], '^', 1);

End {Blank Template form}

else

begin {All other senders}

if Page95Control1.ActivePage = TabReports Then

begin

if ORReportsAvailable.ItemIndex = -1 Then

begin

ErrorText := 'Please select a report type first.';

end

else

footdescrip := 'REPORT: ' + ORReportsAvailable.Items[OrReportsAvailable.ItemIndex];

end

else if Page95Control1.ActivePage = TabDoD then

begin

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 192 or earlier

if ORListBoxDoDReportTypes.ItemIndex = -1 then

begin

ErrorText := 'Please select a report type first.';

end

else

footdescrip := 'FHIE: ' + Piece(ORListBoxDoDReportTypes.Items[ORListBoxDoDReportTypes.ItemIndex], '^', 1);

end // IsPatchInstalled

end

else if (Page95Control1.ActivePage = AdminDocuments) then

begin

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&V) Address') then

footdescrip := 'Address';

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&W) Appointments') Then

footdescrip := lblApptStatus.Caption;

end

else if Page95Control1.ActivePage = TabClinicalDocuments Then

begin {Clinical Documents}

if ((lstDocs.ItemIndex = -1) And (DocType <> 'DIET')) And (DocType <> 'MULTI') Then

begin

ErrorText := 'Please select a clinical document to print.';

end

else

begin {Clinical Document Types}

If DocType = 'PN' Then

footdescrip := 'CLIN DOC: Progress Note'

else If DocType = 'DS' Then

footdescrip := 'CLIN DOC: Discharge Summary'

else If DocType = 'CN' Then

footdescrip := 'CLIN DOC: Consult'

else If DocType = 'VS' Then

footdescrip := 'CLIN DOC: Vital Signs (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'MEDS' Then

footdescrip := 'CLIN DOC: Medications (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'LAB' Then

footdescrip := 'CLIN DOC: Labs (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'IMAGING' Then

footdescrip := 'CLIN DOC: Imaging'

else If DocType = 'DIET' Then

footdescrip := 'CLIN DOC: Diet'

else If DocType = 'NUTASSESS' Then

footdescrip := 'CLIN DOC: Nutritional Assessment (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'ORDERS' Then

footdescrip := 'CLIN DOC: Orders (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'PROCEDURES' Then

footdescrip := 'CLIN DOC: Procedure'

else If DocType = 'PROBLEM LIST' Then

footdescrip := 'CLIN DOC: Problem List (' + lstDocs.Items[lstDocs.ItemIndex] + ')'

else If DocType = 'MULTI' Then

footdescrip := 'CLIN DOC: Multi-Document';

end; {Clinical Document Types}

End {Clinical Documents}

else If Page95Control1.ActivePage = TabHealthSummaries Then

Begin {Health Summaries}

if (ORHealthSummaryUserList.ItemIndex = -1) and

(HSMemoLocal.Lines.Count = 0) Then

begin

ErrorText := 'Please select a health summary type first.';

end

else

begin

If (ORHealthSummaryUserList.ItemIndex > -1) Then

footdescrip := 'HEALTH SUMMARY (' + Tab95Control2.Tabs[Tab95Control2.TabIndex] + '): ' + ORHealthSummaryUserList.Items[OrHealthSummaryUserList.ItemIndex]

Else

footdescrip := 'HEALTH SUMMARY (' + Tab95Control2.Tabs[Tab95Control2.TabIndex] + '): Adhoc Report';

end;

End; {Health Summaries}

End; {All other senders}

if (ErrorText <> '') then

Result := False

else

Result := True;

end;

Table 48: Functions

| Function Name | FormatReportBody | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Remove DoD related report code. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | FormatReportBody | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

{========================================================================

FormatReportBody

This procedure populates the QuickReport memo control with the report

text. Refactored from actFilePrintExecute.

Input: Sender - form

Output: none

//CodeCR353 - rpm 4/30/12

=========================================================================}

procedure TfrmMain.FormatReportBody(Sender: TObject);

var

x, y: Integer;

begin

formPrint.QRMemoReport.ParentRichEdit := Nil;

formPrint.QRMemoReport.Lines.Clear;

formPrint.QRMemoReport.Font.Size := 11; //default font size

if Assigned(Sender) and (Sender = PNCSForm) Then

begin {PNCS Exam report}

formPrint.QRMemoReport.ParentRichEdit := PNCSForm.xSampleReportOutput;

end {PNCS Exam report}

else if Assigned(Sender) and (Sender = FormTIUDisplay) Then

begin {CPRS Document view}

If formTIUDisplay.SpeedButtonCAPRI.Down = True Then

Begin

formPrint.QRMemoReport.ParentRichEdit := formTIUDisplay.MemoDocs;

End

Else

If formTIUDisplay.MemoDocs.Lines.Count > 0 Then

Begin

For y := 0 To formTIUDisplay.MemoDocs.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(formTIUDisplay.MemoDocs.Lines[y]);

End;

End;

end {CPRS Document view}

else If Assigned(Sender) and (Sender = frmExamDetails) Then

Begin {Exam Details form}

If frmExamDetails.FMExamDetailsCommentsMemo.FMField = '70' Then

Begin

// Check for proper font on printer for 132 column printouts

If frmExamDetails.RichEditExamReport.Lines.Count > 0 Then

For y := 0 To frmExamDetails.RichEditExamReport.Lines.Count - 1 Do

Begin

If Length(frmExamDetails.RichEditExamReport.Lines[y]) > 80 Then

Begin

formPrint.QRMemoReport.Font.Size := 7;

End;

End;

//

formPrint.QRMemoReport.Lines.Add('DATE OF EXAM: ' + frmExamDetails.FMDateOfExam.Text);

formPrint.QRMemoReport.Lines.Add('EXAMINING PHYSICIAN: ' + frmExamDetails.FMPhysician.Text);

formPrint.QRMemoReport.Lines.Add('STATUS: ' + frmExamDetails.FMExamStatus.Text);

formPrint.QRMemoReport.Lines.Add(' ');

For y := 0 To frmExamDetails.RichEditExamReport.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(frmExamDetails.RichEditExamReport.Lines[y]);

End;

End;

If frmExamDetails.FMExamDetailsCommentsMemo.FMField = '71' Then

Begin

formPrint.QRMemoReport.ParentRichEdit := frmExamDetails.RichEditExamReport;

frmExamDetails.RichEditExamReport.Lines.Insert(0, ' ');

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'STATUS: ' + frmExamDetails.FMExamStatus.Text);

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'EXAMINING PHYSICIAN: ' + frmExamDetails.FMPhysician.Text);

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'DATE OF EXAM: ' + frmExamDetails.FMDateOfExam.Text);

End;

End {Exam Details form}

else If Assigned(Sender) and (Sender = frmGenerateBlankTemplate) Then

Begin {Blank Template form}

If frmGenerateBlankTemplate.xSampleReportOutput.Lines.Count > 0 Then

For y := 0 To frmGenerateBlankTemplate.xSampleReportOutput.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(frmGenerateBlankTemplate.xSampleReportOutput.Lines[y]);

End;

End {Blank Template form}

else if (Page95Control1.ActivePage = AdminDocuments) then {Admin tab}

begin {Admin Tab}

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&V) Address') then

begin {Address subtab}

formPrint.QRMemoReport.Lines.Add('PATIENT: ' + Panel1.Caption);

formPrint.QRMemoReport.Lines.Add(' ');

formPrint.QRMemoReport.Lines.Add('\*\* PERMANENT ADDRESS \*\*');

formPrint.QRMemoReport.Lines.Add('ADDRESS: ' + fmEditStreet1.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEditStreet2.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEditStreet3.text);

formPrint.QRMemoReport.Lines.Add(' CITY: ' + fmEditCity.text);

formPrint.QRMemoReport.Lines.Add(' STATE: ' + fmEditState.text);

formPrint.QRMemoReport.Lines.Add(' ZIP: ' + fmEditZip.text);

formPrint.QRMemoReport.Lines.Add(' COUNTY: ' + CountyFake.text);

formPrint.QRMemoReport.Lines.Add(' PHONE: ' + fmEditPhone.text);

formPrint.QRMemoReport.Lines.Add('OFFICE PHONE: ' + fmEditOffice.text);

formPrint.QRMemoReport.Lines.Add(' ');

formPrint.QRMemoReport.Lines.Add('\*\* TEMPORARY ADDRESS \*\*');

formPrint.QRMemoReport.Lines.Add('TEMP ADDRESS ACTIVE? ' + fmEdit1.text);

formPrint.QRMemoReport.Lines.Add('START DATE: ' + fmEdit10.text);

formPrint.QRMemoReport.Lines.Add(' STOP DATE: ' + fmEdit11.text);

formPrint.QRMemoReport.Lines.Add('ADDRESS: ' + fmEdit2.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEdit3.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEdit4.text);

formPrint.QRMemoReport.Lines.Add(' CITY: ' + fmEdit5.text);

formPrint.QRMemoReport.Lines.Add(' STATE: ' + fmEdit6.text);

formPrint.QRMemoReport.Lines.Add(' ZIP: ' + fmEdit7.text);

formPrint.QRMemoReport.Lines.Add(' COUNTY: ' + fmEdit8.text);

formPrint.QRMemoReport.Lines.Add(' PHONE: ' + fmEdit9.text);

end {Address subtab}

else if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&W) Appointments') then

begin {Appointments subtab}

for y := 0 to MemoAppointments.Lines.Count - 1 do

begin

formPrint.QRMemoReport.Lines.Add(MemoAppointments.Lines[y]);

end;

end; {Appointments subtab}

end {Admin Tab}

else If Page95Control1.ActivePage = TabClinicalDocuments Then

Begin {Clinical Documents}

For y := 0 To MemoDocs.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(MemoDocs.Lines[y]);

End;

End {Clinical Documents}

else If Page95Control1.ActivePage = TabReports Then

Begin {Reports Tab}

If ORReportsAvailable.Items[ORReportsAvailable.ItemIndex] = 'C&P Exam Detail' Then

Begin

For y := 0 To ReportMemo.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(ReportMemo.Lines[y]);

End;

End

Else

For y := 0 To ReportMemo.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(ReportMemo.Lines[y]);

End;

End {Reports Tab}

else If Page95Control1.ActivePage = TabDoD Then

Begin {DoD Tab}

For y := 0 To RichEditDODReport.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(RichEditDODReport.Lines[y]);

End;

End {DoD Tab}

else If Page95Control1.ActivePage = TabHealthSummaries Then

Begin {Health Summaries}

TempHSMemo.Clear;

// Determine local or remote health summary

If Tab95Control2.TabIndex = 0 Then

Begin

If HSMemoLocal.Lines.Count > 0 Then

For x := 0 To HSMemoLocal.Lines.Count - 1 Do

TempHSMemo.Add(removetrailingspaces(HSMemoLocal.Lines[x]));

End

Else

Begin

If HSMemo.Lines.Count > 0 Then

For x := 0 To HSMemo.Lines.Count - 1 Do

TempHSMemo.Add(removetrailingspaces(HSMemo.Lines[x]));

End;

For y := 0 To TempHSMemo.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(TempHSMemo[y]);

End;

End {Health Summaries}

else If Assigned(Sender) and (Sender = TIUSignForm) Then // CodeCR423 JRL 11/13/12

Begin {TIUSign Form} // CodeCR423 JRL 11/13/12

For y := 0 To TIUSignForm.TIUtoVVAProgressNotes.Count - 1 Do // CodeCR423 JRL 11/13/12

Begin // CodeCR423 JRL 11/13/12

formPrint.QRMemoReport.Lines.Add(TIUSignForm.TIUtoVVAProgressNotes[y]); // 11/13/12

End; // CodeCR423 JRL 11/13/12

End; {TIUSign Form} // CodeCR423 JRL 11/13/12

end;

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

{========================================================================

FormatReportBody

This procedure populates the QuickReport memo control with the report

text. Refactored from actFilePrintExecute.

Input: Sender - form

Output: none

//CodeCR353 - rpm 4/30/12

=========================================================================}

procedure TfrmMain.FormatReportBody(Sender: TObject);

var

x, y: Integer;

begin

formPrint.QRMemoReport.ParentRichEdit := Nil;

formPrint.QRMemoReport.Lines.Clear;

formPrint.QRMemoReport.Font.Size := 11; //default font size

if Assigned(Sender) and (Sender = PNCSForm) Then

begin {PNCS Exam report}

formPrint.QRMemoReport.ParentRichEdit := PNCSForm.xSampleReportOutput;

end {PNCS Exam report}

else if Assigned(Sender) and (Sender = FormTIUDisplay) Then

begin {CPRS Document view}

If formTIUDisplay.SpeedButtonCAPRI.Down = True Then

Begin

formPrint.QRMemoReport.ParentRichEdit := formTIUDisplay.MemoDocs;

End

Else

If formTIUDisplay.MemoDocs.Lines.Count > 0 Then

Begin

For y := 0 To formTIUDisplay.MemoDocs.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(formTIUDisplay.MemoDocs.Lines[y]);

End;

End;

end {CPRS Document view}

else If Assigned(Sender) and (Sender = frmExamDetails) Then

Begin {Exam Details form}

If frmExamDetails.FMExamDetailsCommentsMemo.FMField = '70' Then

Begin

// Check for proper font on printer for 132 column printouts

If frmExamDetails.RichEditExamReport.Lines.Count > 0 Then

For y := 0 To frmExamDetails.RichEditExamReport.Lines.Count - 1 Do

Begin

If Length(frmExamDetails.RichEditExamReport.Lines[y]) > 80 Then

Begin

formPrint.QRMemoReport.Font.Size := 7;

End;

End;

//

formPrint.QRMemoReport.Lines.Add('DATE OF EXAM: ' + frmExamDetails.FMDateOfExam.Text);

formPrint.QRMemoReport.Lines.Add('EXAMINING PHYSICIAN: ' + frmExamDetails.FMPhysician.Text);

formPrint.QRMemoReport.Lines.Add('STATUS: ' + frmExamDetails.FMExamStatus.Text);

formPrint.QRMemoReport.Lines.Add(' ');

For y := 0 To frmExamDetails.RichEditExamReport.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(frmExamDetails.RichEditExamReport.Lines[y]);

End;

End;

If frmExamDetails.FMExamDetailsCommentsMemo.FMField = '71' Then

Begin

formPrint.QRMemoReport.ParentRichEdit := frmExamDetails.RichEditExamReport;

frmExamDetails.RichEditExamReport.Lines.Insert(0, ' ');

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'STATUS: ' + frmExamDetails.FMExamStatus.Text);

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'EXAMINING PHYSICIAN: ' + frmExamDetails.FMPhysician.Text);

frmExamDetails.RichEditExamReport.Lines.Insert(0, 'DATE OF EXAM: ' + frmExamDetails.FMDateOfExam.Text);

End;

End {Exam Details form}

else If Assigned(Sender) and (Sender = frmGenerateBlankTemplate) Then

Begin {Blank Template form}

If frmGenerateBlankTemplate.xSampleReportOutput.Lines.Count > 0 Then

For y := 0 To frmGenerateBlankTemplate.xSampleReportOutput.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(frmGenerateBlankTemplate.xSampleReportOutput.Lines[y]);

End;

End {Blank Template form}

else if (Page95Control1.ActivePage = AdminDocuments) then {Admin tab}

begin {Admin Tab}

if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&V) Address') then

begin {Address subtab}

formPrint.QRMemoReport.Lines.Add('PATIENT: ' + Panel1.Caption);

formPrint.QRMemoReport.Lines.Add(' ');

formPrint.QRMemoReport.Lines.Add('\*\* PERMANENT ADDRESS \*\*');

formPrint.QRMemoReport.Lines.Add('ADDRESS: ' + fmEditStreet1.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEditStreet2.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEditStreet3.text);

formPrint.QRMemoReport.Lines.Add(' CITY: ' + fmEditCity.text);

formPrint.QRMemoReport.Lines.Add(' STATE: ' + fmEditState.text);

formPrint.QRMemoReport.Lines.Add(' ZIP: ' + fmEditZip.text);

formPrint.QRMemoReport.Lines.Add(' COUNTY: ' + CountyFake.text);

formPrint.QRMemoReport.Lines.Add(' PHONE: ' + fmEditPhone.text);

formPrint.QRMemoReport.Lines.Add('OFFICE PHONE: ' + fmEditOffice.text);

formPrint.QRMemoReport.Lines.Add(' ');

formPrint.QRMemoReport.Lines.Add('\*\* TEMPORARY ADDRESS \*\*');

formPrint.QRMemoReport.Lines.Add('TEMP ADDRESS ACTIVE? ' + fmEdit1.text);

formPrint.QRMemoReport.Lines.Add('START DATE: ' + fmEdit10.text);

formPrint.QRMemoReport.Lines.Add(' STOP DATE: ' + fmEdit11.text);

formPrint.QRMemoReport.Lines.Add('ADDRESS: ' + fmEdit2.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEdit3.text);

formPrint.QRMemoReport.Lines.Add(' ' + fmEdit4.text);

formPrint.QRMemoReport.Lines.Add(' CITY: ' + fmEdit5.text);

formPrint.QRMemoReport.Lines.Add(' STATE: ' + fmEdit6.text);

formPrint.QRMemoReport.Lines.Add(' ZIP: ' + fmEdit7.text);

formPrint.QRMemoReport.Lines.Add(' COUNTY: ' + fmEdit8.text);

formPrint.QRMemoReport.Lines.Add(' PHONE: ' + fmEdit9.text);

end {Address subtab}

else if (Tab95Control3.Tabs[Tab95Control3.TabIndex] = '&W) Appointments') then

begin {Appointments subtab}

for y := 0 to MemoAppointments.Lines.Count - 1 do

begin

formPrint.QRMemoReport.Lines.Add(MemoAppointments.Lines[y]);

end;

end; {Appointments subtab}

end {Admin Tab}

else If Page95Control1.ActivePage = TabClinicalDocuments Then

Begin {Clinical Documents}

For y := 0 To MemoDocs.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(MemoDocs.Lines[y]);

End;

End {Clinical Documents}

else If Page95Control1.ActivePage = TabReports Then

Begin {Reports Tab}

If ORReportsAvailable.Items[ORReportsAvailable.ItemIndex] = 'C&P Exam Detail' Then

Begin

For y := 0 To ReportMemo.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(ReportMemo.Lines[y]);

End;

End

Else

For y := 0 To ReportMemo.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(ReportMemo.Lines[y]);

End;

End {Reports Tab}

else If Page95Control1.ActivePage = TabDoD Then

Begin

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 192 and earlier {DoD Tab}

For y := 0 To RichEditDODReport.Lines.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(RichEditDODReport.Lines[y]);

End;

End;

End {DoD Tab}

else If Page95Control1.ActivePage = TabHealthSummaries Then

Begin {Health Summaries}

TempHSMemo.Clear;

// Determine local or remote health summary

If Tab95Control2.TabIndex = 0 Then

Begin

If HSMemoLocal.Lines.Count > 0 Then

For x := 0 To HSMemoLocal.Lines.Count - 1 Do

TempHSMemo.Add(removetrailingspaces(HSMemoLocal.Lines[x]));

End

Else

Begin

If HSMemo.Lines.Count > 0 Then

For x := 0 To HSMemo.Lines.Count - 1 Do

TempHSMemo.Add(removetrailingspaces(HSMemo.Lines[x]));

End;

For y := 0 To TempHSMemo.Count - 1 Do

Begin

formPrint.QRMemoReport.Lines.Add(TempHSMemo[y]);

End;

End {Health Summaries}

else If Assigned(Sender) and (Sender = TIUSignForm) Then // CodeCR423 JRL 11/13/12

Begin {TIUSign Form} // CodeCR423 JRL 11/13/12

For y := 0 To TIUSignForm.TIUtoVVAProgressNotes.Count - 1 Do // CodeCR423 JRL 11/13/12

Begin // CodeCR423 JRL 11/13/12

formPrint.QRMemoReport.Lines.Add(TIUSignForm.TIUtoVVAProgressNotes[y]); // 11/13/12

End; // CodeCR423 JRL 11/13/12

End; {TIUSign Form} // CodeCR423 JRL 11/13/12

end;

Table 48: Functions

| Function Name | actFileTransmitVirtualVAExecute | | | |
| --- | --- | --- | --- | --- |
| **Short Description** | Make the code that sends DoD documents to Virtual VA patch sensitive. Only used for Patch 192 and earlier. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | N/A | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | N/A | N/A |

| Function Name | actFileTransmitVirtualVAExecute | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **Data Passing** | Input | Output | Both | Global Reference | Local Reference |
| **Input Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |
| **Output Attribute Name and Definition** | Name: N/A | | | | |
| Definition: | | | | |

| Current Logic |
| --- |

{========================================================================

actFileTransmitVirtualVAExecute

This action handler processes text into a PDF file and transmits the

PDF file to the Virtual VA claims systems at the VBA.

CodeCR353 - rpm 5/1/12

========================================================================}

procedure TfrmMain.actFileTransmitVirtualVAExecute(Sender: TObject);

var

DtTm: String;

currentDivision, localStation: String;

footDescrip: String;

SendVVA: TfrmVirtualVA;

fileSizeBytes: LongInt;

SendFile: Boolean;

buffer: string;

const

MAX\_SIZE = 15728640; //The maximum file size in bytes to prevent exceeding 15 mb files.

DOC\_TYPE = 344; //CAPRI

DOC\_TYPE\_CP = 356; //C&P Exams // CodeCR539 JRL 09/27/13

DOC\_CATEGORY = 44; //Medical Records

DOC\_SUBJECT = 'Clinical Documents';

DOC\_SOURCE = 'CAPRIVISTA';

DOC\_TYPE\_DOD = 89; // DoD Tab // CodeCR425 JRL 10/19/12

DOC\_CATEGORY\_DOD = 51; // DoD Category // CodeCR425 JRL 10/19/12

DOC\_SOURCE\_COMMENT\_DOD = 'BHIE FHIE'; // CodeCR425 JRL 11/13/12

DOC\_DOD\_SUBJECT = 'DoD Documents'; // CodeCR425 JRL 01/0/12

begin

VVARetransmit := TRUE; // CodeCR710 JRL 6/11/15

while VVARetransmit do // CodeCR710 JRL 6/11/15

begin // CodeCR710 JRL 6/11/15

VVARetransmit := FALSE; // CodeCR710 JRL 6/11/15

DtTm := '';

currentDivision := '';

localStation := '';

footdescrip := '';

SendVVA := TfrmVirtualVA.Create(self);

// SendVVA.Position := poOwnerFormCenter;

SendVVA.Position := poScreenCenter; // rra I8233197FY16 Patch 193 7/20/16

//do we want to delete the temporary PDF file after transmission?

SendVVA.vvaWSC.IsDeleteFile := NOT FNoDeleteVVA;

try

if FormatReport(Sender, footdescrip, DtTm, currentDivision) then

try {ReportOK}

if (formprint.QRMemoReport.Lines.Count > 0) then

begin {Report Text available}

SendVVA.vvaWSC.LoadText(formPrint.QRMemoReport);

//Create PDF file

with SendVVA.vvaWSC.VVAReport do

begin

FooterDesc := footDescrip;

if (formPRint.QRMemoReport.Font.Size = 7) then

FontSize := 7

else

FontSize := 10;

FontName := 'Courier New';

PrintDate := DtTm;

SystemName := Uppercase(RPCBroker1.ANUStrServer);

Division := currentDivision;

Patient.FullName := patientName;

Patient.SSN := patientSSN;

end;

fileSizeBytes := SendVVA.vvaWSC.CreateXmitFile(SendVVA.vvaWSC.vvaReport);

// copy the file to the heap for vler das use

if fileSizeBytes > 0 then

begin

PdfMemoryStream := TMemoryStream.Create;

try

if Sender is TForm then // CodeCR565 JRL 07/31/14

begin // CodeCR565 JRL 07/31/14

if (Sender as TForm).Name = 'TIUSignForm' then // CodeCR565 JRL 07/31/14

begin // CodeCR565 JRL 07/31/14

TIUSignForm.FinalReport := TMemoryStream.Create; // CodeCR565 JRL 7/20/14

TIUSignForm.FinalReport.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + 'c' + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName); // CodeCR565 JRL 7/20/14 (Save report for cosigned exams)

end;

end; // CodeCR538 JRL 07/31/14

//// PdfMemoryStream.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName);

// Read compressed PDF into memory to send to VLER/DAS.

PdfMemoryStream.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + 'c' + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName); // CodeCR559 JRL 3/17/14

// Save final report - it may be needed for cosignature report

except

on E: Exception do

begin

MessageDlg('PDF report could not be copied and will NOT be sent to VLER.', mtError, [mbOK], 0);

FreeAndNil(PdfMemoryStream);

end;

end;

end;

(\* The requirements limit Virtual VA PDF format files to 15 MB. \*)

if (fileSizeBytes < MAX\_SIZE) then

begin {Does not exceed size limit}

// CodeCR538: If coming from TIUSign form, don't ask for confirmation

// for the automatic transmission to VVA. Coming from any other form

// or control is manual so confirm sending

SendFile := FALSE; // CodeCR538 JRL 11/18/13

if Sender is TForm then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/18/13

if (Sender as TForm).Name = 'TIUSignForm' then // CodeCR538 JRL 11/18/13

SendFile := TRUE // CodeCR538 JRL 11/18/13

end; // CodeCR538 JRL 11/18/13

if SendFile = FALSE then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/18/13

if SendVVA.vvaWSC.IsConfirmedToSend(patientName) then // CodeCR538 JRL 11/18/13

SendFile := TRUE // CodeCR538 JRL 11/18/13

else // CodeCR538 JRL 11/18/13

SendFile := FALSE // CodeCR538 JRL 11/18/13

end; // CodeCR538 JRL 11/18/13

  if SendFile then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/15/13

{Get Local Station name and number}

localStation := GetDivision(False, RPCBroker1);

{Populate fields required to transmit to the Virtual VA web

service}

with SendVVA.vvaWSC.VVAReport do

begin

VVADoc.DocCategory := DOC\_CATEGORY;

VVADoc.Source := DOC\_SOURCE;

buffer := Piece(localStation, '^', 2) + ' (' + Piece(localStation, '^', 3) + ')';

VVADoc.SourceComment := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

VVADoc.User := RPCBroker1.User.Name;

VVADoc.URL := FVirtualVAURL;

VVADoc.SecToken := FVirtualVAToken;

{Use C&P Document type and Exam as subject for C&P Exams}

if (Sender = frmExamDetails) then

begin

VVADoc.DocType := DOC\_TYPE\_CP;

buffer := frmExamDetails.FMExamType.Text;

VVADoc.Subject := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

End

else if Page95Control1.ActivePage = TabDOD then // CodeCR425 JRL 10/19/12

begin // CodeCR425 JRL 10/19/12

VVADoc.DocType := DOC\_TYPE\_DOD; // CodeCR425 JRL 10/19/12

VVADoc.DocCategory := DOC\_CATEGORY\_DOD; // CodeCR425 JRL 10/19/12

VVADoc.Subject := DOC\_DOD\_SUBJECT; // CodeCR425 JRL 10/19/12

VVADoc.SourceComment := DOC\_SOURCE\_COMMENT\_DOD; // CodeCR425 JRL 11/9/12

end // CodeCR425 JRL 10/19/12

else if (Sender = TIUSignForm) then // CodeCR539 JRL 09/27/13 //C&P Exam Title

begin // CodeCR539 JRL 09/27/13

VVADoc.DocType := DOC\_TYPE\_CP; // CodeCR539 JRL 09/27/13

buffer := TIUSignForm.ParentExamTitle; // CodeCR522 JRL 11/21/13

VVADoc.Subject := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

end // CodeCR539 JRL 09/27/13

else {Default to CAPRI (344) and 'Clinical Documents'}

begin

VVADoc.DocType := DOC\_TYPE;

VVADoc.Subject := DOC\_SUBJECT;

end;

end;

if VAUtils.ScreenReaderActive then //508 compliance

SendVVA.ActiveControl := SendVVA.edVirtualVATitle;

SendVVA.ShowModal;

end; {OKToSend} // CodeCR538 JRL 11/15/13

end {Does not exceed line count limit}

else {Exceeds line count limit}

begin

ShowMessageCAPRI('The document file contains '

+ SendVVA.vvaWSC.FormatByteSize(fileSizeBytes)

+ ', which exceeds the 15 MB'

+ ' maximum file size allowed.');

end;

end {Report Text available}

else {NO Report Text available}

begin

ShowMessageCAPRI('No text to send to Virtual VA.');

end;

finally

CleanupAfterReport(Sender);

end; {ReportOK}

finally

FreeAndNil(SendVVA);

AnimateLogo(False); // rra 998063 change status back to ready

StatusBarLoadPt.Caption := 'Ready.'; // rra 998063

StatusBarLoadPt.Repaint; // rra 998063

Application.Processmessages; // rra 998063

end;

end; // while VVARetransmit // CodeCR710 JRL 6/11/15

end;

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

{========================================================================

actFileTransmitVirtualVAExecute

This action handler processes text into a PDF file and transmits the

PDF file to the Virtual VA claims systems at the VBA.

CodeCR353 - rpm 5/1/12

========================================================================}

procedure TfrmMain.actFileTransmitVirtualVAExecute(Sender: TObject);

var

DtTm: String;

currentDivision, localStation: String;

footDescrip: String;

SendVVA: TfrmVirtualVA;

fileSizeBytes: LongInt;

SendFile: Boolean;

buffer: string;

const

MAX\_SIZE = 15728640; //The maximum file size in bytes to prevent exceeding 15 mb files.

DOC\_TYPE = 344; //CAPRI

DOC\_TYPE\_CP = 356; //C&P Exams // CodeCR539 JRL 09/27/13

DOC\_CATEGORY = 44; //Medical Records

DOC\_SUBJECT = 'Clinical Documents';

DOC\_SOURCE = 'CAPRIVISTA';

DOC\_TYPE\_DOD = 89; // DoD Tab // CodeCR425 JRL 10/19/12 // DoD Category // CodeCR425 JRL 10/19/12

DOC\_SOURCE\_COMMENT\_DOD = 'BHIE FHIE'; // CodeCR425 JRL 11/13/12

DOC\_DOD\_SUBJECT = 'DoD Documents'; // CodeCR425 JRL 01/0/12

begin

VVARetransmit := TRUE; // CodeCR710 JRL 6/11/15

while VVARetransmit do // CodeCR710 JRL 6/11/15

begin // CodeCR710 JRL 6/11/15

VVARetransmit := FALSE; // CodeCR710 JRL 6/11/15

DtTm := '';

currentDivision := '';

localStation := '';

footdescrip := '';

SendVVA := TfrmVirtualVA.Create(self);

// SendVVA.Position := poOwnerFormCenter;

SendVVA.Position := poScreenCenter; // rra I8233197FY16 Patch 193 7/20/16

//do we want to delete the temporary PDF file after transmission?

SendVVA.vvaWSC.IsDeleteFile := NOT FNoDeleteVVA;

try

if FormatReport(Sender, footdescrip, DtTm, currentDivision) then

try {ReportOK}

if (formprint.QRMemoReport.Lines.Count > 0) then

begin {Report Text available}

SendVVA.vvaWSC.LoadText(formPrint.QRMemoReport);

//Create PDF file

with SendVVA.vvaWSC.VVAReport do

begin

FooterDesc := footDescrip;

if (formPRint.QRMemoReport.Font.Size = 7) then

FontSize := 7

else

FontSize := 10;

FontName := 'Courier New';

PrintDate := DtTm;

SystemName := Uppercase(RPCBroker1.ANUStrServer);

Division := currentDivision;

Patient.FullName := patientName;

Patient.SSN := patientSSN;

end;

fileSizeBytes := SendVVA.vvaWSC.CreateXmitFile(SendVVA.vvaWSC.vvaReport);

// copy the file to the heap for vler das use

if fileSizeBytes > 0 then

begin

PdfMemoryStream := TMemoryStream.Create;

try

if Sender is TForm then // CodeCR565 JRL 07/31/14

begin // CodeCR565 JRL 07/31/14

if (Sender as TForm).Name = 'TIUSignForm' then // CodeCR565 JRL 07/31/14

begin // CodeCR565 JRL 07/31/14

TIUSignForm.FinalReport := TMemoryStream.Create; // CodeCR565 JRL 7/20/14

TIUSignForm.FinalReport.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + 'c' + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName); // CodeCR565 JRL 7/20/14 (Save report for cosigned exams)

end;

end; // CodeCR538 JRL 07/31/14

//// PdfMemoryStream.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName);

// Read compressed PDF into memory to send to VLER/DAS.

PdfMemoryStream.LoadFromFile(SendVVA.vvaWSC.vvaReport.VVADoc.OutputFilePath + 'c' + SendVVA.vvaWSC.vvaReport.VVADoc.OutputFileName); // CodeCR559 JRL 3/17/14

// Save final report - it may be needed for cosignature report

except

on E: Exception do

begin

MessageDlg('PDF report could not be copied and will NOT be sent to VLER.', mtError, [mbOK], 0);

FreeAndNil(PdfMemoryStream);

end;

end;

end;

(\* The requirements limit Virtual VA PDF format files to 15 MB. \*)

if (fileSizeBytes < MAX\_SIZE) then

begin {Does not exceed size limit}

// CodeCR538: If coming from TIUSign form, don't ask for confirmation

// for the automatic transmission to VVA. Coming from any other form

// or control is manual so confirm sending

SendFile := FALSE; // CodeCR538 JRL 11/18/13

if Sender is TForm then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/18/13

if (Sender as TForm).Name = 'TIUSignForm' then // CodeCR538 JRL 11/18/13

SendFile := TRUE // CodeCR538 JRL 11/18/13

end; // CodeCR538 JRL 11/18/13

  if SendFile = FALSE then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/18/13

if SendVVA.vvaWSC.IsConfirmedToSend(patientName) then // CodeCR538 JRL 11/18/13

SendFile := TRUE // CodeCR538 JRL 11/18/13

else // CodeCR538 JRL 11/18/13

SendFile := FALSE // CodeCR538 JRL 11/18/13

end; // CodeCR538 JRL 11/18/13

if SendFile then // CodeCR538 JRL 11/18/13

begin // CodeCR538 JRL 11/15/13

{Get Local Station name and number}

localStation := GetDivision(False, RPCBroker1);

{Populate fields required to transmit to the Virtual VA web

service}

with SendVVA.vvaWSC.VVAReport do

begin

VVADoc.DocCategory := DOC\_CATEGORY;

VVADoc.Source := DOC\_SOURCE;

buffer := Piece(localStation, '^', 2) + ' (' + Piece(localStation, '^', 3) + ')';

VVADoc.SourceComment := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

VVADoc.User := RPCBroker1.User.Name;

VVADoc.URL := FVirtualVAURL;

VVADoc.SecToken := FVirtualVAToken;

{Use C&P Document type and Exam as subject for C&P Exams}

if (Sender = frmExamDetails) then

begin

VVADoc.DocType := DOC\_TYPE\_CP;

buffer := frmExamDetails.FMExamType.Text;

VVADoc.Subject := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

end

else if Page95Control1.ActivePage = TabDOD then // CodeCR425 JRL 10/19/12

begin

// Patch 193 - Remove DoD tab and associated code once Patch 193 is installed JRL 7/20/16

If not IsPatchInstalled('DVBA\*2.7\*193') then

begin // Patch 192 and earlier // CodeCR425 JRL 10/19/12

VVADoc.DocType := DOC\_TYPE\_DOD; // CodeCR425 JRL 10/19/12

VVADoc.DocCategory := DOC\_CATEGORY\_DOD; // CodeCR425 JRL 10/19/12

VVADoc.Subject := DOC\_DOD\_SUBJECT; // CodeCR425 JRL 10/19/12

VVADoc.SourceComment := DOC\_SOURCE\_COMMENT\_DOD; // CodeCR425 JRL 11/9/12

end;

end // CodeCR425 JRL 10/19/12

else if (Sender = TIUSignForm) then // CodeCR539 JRL 09/27/13 C&P Exam Title // CodeCR539 JRL 09/27/13

VVADoc.DocType := DOC\_TYPE\_CP; // CodeCR539 JRL 09/27/13

buffer := TIUSignForm.ParentExamTitle; // CodeCR522 JRL 11/21/13

VVADoc.Subject := RemoveSpecialCharacters(buffer); // CodeCR522 JRL 11/21/13

end // CodeCR539 JRL 09/27/13

else {Default to CAPRI (344) and 'Clinical Documents'}

begin

VVADoc.DocType := DOC\_TYPE;

VVADoc.Subject := DOC\_SUBJECT;

end;

end;

if VAUtils.ScreenReaderActive then //508 compliance

SendVVA.ActiveControl := SendVVA.edVirtualVATitle;

SendVVA.ShowModal;

end; {OKToSend} // CodeCR538 JRL 11/15/13

end {Does not exceed line count limit}

else {Exceeds line count limit}

begin

ShowMessageCAPRI('The document file contains '

+ SendVVA.vvaWSC.FormatByteSize(fileSizeBytes)

+ ', which exceeds the 15 MB'

+ ' maximum file size allowed.');

end;

end {Report Text available}

else {NO Report Text available}

begin

ShowMessageCAPRI('No text to send to Virtual VA.');

end;

finally

CleanupAfterReport(Sender);

end; {ReportOK}

finally

FreeAndNil(SendVVA);

AnimateLogo(False); // rra 998063 change status back to ready

StatusBarLoadPt.Caption := 'Ready.'; // rra 998063

StatusBarLoadPt.Repaint; // rra 998063

Application.Processmessages; // rra 998063

end;

end; // while VVARetransmit // CodeCR710 JRL 6/11/15

end;

##### Dialog – N/A for Requirement 2.6.2

##### Help Frame – N/A for Requirement 2.6.2

##### HL7 Application Parameter – N/A for Requirement 2.6.2

##### HL7 Logical Link – N/A for Requirement 2.6.2

##### COTS Interface – N/A for Requirement 2.6.2

## Network Detailed Design

CAPRI GUI communicates with various servers within the VA intranet via TCP/IP. Existing VA intranet WAN backbone is leveraged as needed by any software component or service within the application. Type of traffic over TCP/IP includes, RPCBroker, HTTP and HTTPS.

## Security and Privacy

### Security

A System Security Plan is not required.  As a Delphi-based GUI front-end for the VistA AMIE package, CAPRI adheres to all VA Information Technology System Policies.  CAPRI implements VistA KERNEL information security methodology for user access and identity management, auditing,  record restriction and menu availability.  It also uses RPC Broker’s Broker Security Enhancement features, to access the underlying AMIE package functionality.  In addition to using established mechanisms to ensure only authorized access to medical data, CAPRI adds a level of security by allowing VBA users to read but not alter electronic medical record information.

CAPRI’s Authority To Operate (ATO) can be found at the following link:

[http://DNS DNS .DNS /development/legacy\_product\_enhancements/healthcare\_management\_products/CAPRI/CAPRI%20Projects/ATO/Compensation%20and%20Pension%20Record%20Interchange%20(CAPRI)%20ATO.pdf](http://DNS  DNS       .DNS   /development/legacy_product_enhancements/healthcare_management_products/CAPRI/CAPRI%20Projects/ATO/Compensation%20and%20Pension%20Record%20Interchange%20%28CAPRI%29%20ATO.pdf)

#### Security Control Families

CAPRI adheres to the VA Information Technology System Policies.

### Security Management

VA Directive 10-93-142 prohibits local modifications to VistA software.

### General Security

For general security info for CAPRI, AMIE and VistA, see the CAPRI GUI User Manual located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below. This manual includes instructions for setting up CAPRI users, as well as descriptions of all Security Keys used by the CAPRI GUI application. The direct link is below.

[http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

### Remote Systems

The AMIE software does not transmit data to any remote systems. For CAPRI interactions with remote systems, refer to the Systems Architecture diagram in Section 4.

### Contingency Planning

Your facility should have a local contingency plan in the event of application problems in a live environment. It should identify the procedure for maintaining functionality provided by the AMIE software as well as the CAPRI GUI application, in the event of system outage.

### Interfacing

There are no special interfacing requirements for the AMIE or the CAPRI software.

### Electronic Signatures

The CAPRI GUI application uses electronic signatures. Use the following link to locate the CAPRI GUI User Manual: [http://www.DNS /vdl/application.asp?appid=133](http://www.DNS   /vdl/application.asp?appid=133)

### Security Keys

Take the following steps to get information about the security keys used with the AMIE software.

1. VA FileMan Menu

2. Print File Entries Option

3. Output from what File: SECURITY KEY

4. Sort by: Name

5. Start with name: DVBA to DVBC

6. Within name, sort by: <RET>

7. First print field: Name

8. Then print field: Description

***Note:*** *Some keys do not affect the menu operation. This is due to some options having several different functions which are limited in scope by the key. This limitation is done internally by the program being used.*

### Privacy

Privacy is maintained through system access restrictions based on [VA DIRECTIVE 6500 InformationSecurity](http://www.DNS   /vapubs/viewPublication.asp?Pub_ID=374&FType=2)

## Service Oriented Architecture / ESS Detailed Design

### Service Description for Virtual VA

Virtual VA Paperless Claims Processing is a Web-based suite of information solutions that provides electronic "eFolders" for claims processing through imaging, document management technologies, and integration with output capabilities of several other VA systems. Full implementation of Virtual VA enables VBA to address timely processing of services to veterans, which is currently hindered by the

dependence on paper documents. Full implementation will also greatly improve the security and the privacy of veteran data by migrating from a vulnerable, paper claims folder to a secure, centralized, and electronic claims folder for compensation and pension processing. Electronic document workflow processing, business process management, VA and DoD systems integration,national scanning of documents, and improved workflow management. Currently, Virtual VA is designed to provide an electronic document management system for Compensation and Pension (C&P) Service

and only for pension claims. Due to Congressional and Presidential interest in providing faster service to veterans, VA is proposing that Virtual VA be expanded to support paperless claims processing for all C&P activities. The web-based nature of the system will allow increased accessibility and information sharing across the 58 regional offices / external organizations. Virtual VA is accessed via a SOAP interface.

### Service Design for Virtual VA

Virtual VA is accessed via a SOAP interface. Virtual VA-2009. [http://www.DNS /EXHIBIT300/docs/VV.pdf](http://www.DNS   /EXHIBIT300/docs/VV.pdf)

### Service Description for VLER/DAS

This capability with the eHealth Exchange renders VistAWeb a key component of Virtual Lifetime Electronic Record (VLER) electronic health information exchange. Uses Object Oriented Programming to access Vista Pateint data.

### Service Design for Virtual VA

VLER/DAS web service requires two-way SSL (also known as mutual authentication). CAPRI has been modified to relay VLER DAS web service requests through the CAPRI Proxy service using one-way SSL.

### Service Description for VistA Web Server

Veterans Health Information Systems and Technology Architecture (VistA) [https://vistaweb.med.DNS /](https://vistaweb.med.DNS   /) is an intranet web application used to review remote patient information found in VistA, the Bi-Directional Health Information Exchange (BHIE) system, the Health Data Repository II (HDR II) databases, and the Nationwide Health Information Network (NwHIN).

### Service Design for VistA Web Server

VistAWeb, Version 16.1 (Patch WEBV\*1\*26)

### Service Description for HealtheVet Web Services Client

As VistA is migrated to the new Health*e*Vet-VistA system, some legacy VistA applications will need synchronous access to Health*e*Vet application services and data. HWSC uses Caché’s Web Services Client to invoke web service methods on external servers and retrieve results. It provides helper methods and classes to improve the use of the Web Service Client in a Health*e*Vet-VistA environment

### Service Design for HealtheVet Web Services Client

HWSC acts as an adjunct to the web services client functionality provided in Caché, by:

* Leveraging Caché's platform-provided Web services client capabilities.
* Adding a file and UI to manage the set of external web server endpoints (IP, port, etc.)
* Adding a file and UI to register and manage the set of external web services.
* Providing runtime API to invoke a specific web service on a specific web server.
* Providing a runtime API to facilitate error processing in a VistA environment.
* Providing a deployment API to install/register a web service proxy from a WSDL file.
* Providing a management UI including the ability to ‘ping’ (test) a given web service/server combination from VistA/M.
* Supporting both SOAP- and REST-style web services
* Fostering consistent implementation of VistA/M web service consumers

# External System Interface Design

This section details interfaces external to system, that are NOT services (ESS/SOA). Typically, these may include, RPCs, Flat Data Files etc.

## Interface Architecture

Figure 7 - 1 – Interface Architecture



Table 7-1 – Interface Architecture Descriptions

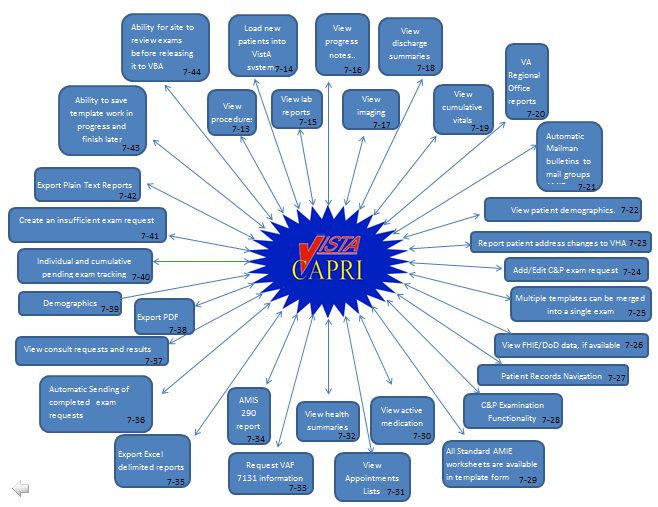
| ID | Name | Description | Interface type | Specifications |
| --- | --- | --- | --- | --- |
| 7-1 | LOCAL VAMC  VISTA(CACHE) FILEMAN AND  REMOTE PROCEDURE CALL/BROKER | **InterSystems Caché** is a commercial object-oriented database management system from InterSystems, used to develop healthcare management and telecommunications software. Customer software can use the database with object and SQL code. Caché also allows developers to directly manipulate its underlying data structures: hierarchical arrays known as M technology. **FileMan** is a set of utilities to provide a meta-data function for MUMPS applications. The FileMan utilities allow the definition of data structures, menus and security, reports, and forms, allowing someone to set up applications without tremendous experience in the MUMPS programming language.  Remote Procedure Calls are a set of fileman instructions to bridge the gap between the GUI and VISTA. The RPC Broker is the foundation of a powerful set of RAD Delphi and Java components. | Data Base Management | Cache20111 OpenVMS,Linux  Kernel 8.0 is required to run the RPC Broker and appropriate VISTA Sofware patch updates. |
| 7-2 | VISTA WEB SERVER | Veterans Health Information Systems and Technology Architecture (VistA) [https://vistaweb.med.DNS /](https://vistaweb.med.DNS   /) is an intranet web application used to review remote patient information found in VistA, the Bi-Directional Health Information Exchange (BHIE) system, the Health Data Repository II (HDR II) databases, and the Nationwide Health Information Network (NwHIN). | Web Server | VistAWeb  *Version 16.1*  *(Patch WEBV\*1\*26)* |
| 7-3 | VIRTUAL VA | Virtual VA Paperless Claims Processing is a Web-based suite of information solutions that provides electronic "eFolders" for claims  processing through imaging, document management technologies, and integration with output capabilities of several other VA systems.  Full implementation of Virtual VA enables VBA to address timely processing of services to veterans, which is currently hindered by the  dependence on paper documents. Full implementation will also greatly improve the security and the privacy of veteran data by migrating  from a vulnerable, paper claims folder to a secure, centralized, and electronic claims folder for compensation and pension processing.  Electronic document workflow processing, business process management, VA and DoD systems integration,national scanning of documents, and improved workflow management.  Currently, Virtual VA is designed to provide an electronic document management system for Compensation and Pension (C&P) Service  and only for pension claims. Due to Congressional and Presidential interest in providing faster service to veterans, VA is proposing that  Virtual VA be expanded to support paperless claims processing for all C&P activities. The web-based nature of the system will allow increased accessibility and  information sharing across the 58 regional offices / external organizations. | Virtual **VA** Web Service | Virtual VA-2009  [http://www.DNS /EXHIBIT300/docs/VV.pdf](http://www.DNS   /EXHIBIT300/docs/VV.pdf) |
| 7-4 | VETERANS BENEFIT MANAGEMENT SYSTEM | VBMS is an IT system designed to help claims adjudicators reach timely and informed decisions in a digital environment. VBMS will enable the five steps of the standard claims adjudication process including establishment, development, evidence, rating, and awards | Enterprise data services | External and inter-departmental communications, e.g. VRM, VLER  Technology platform based on Services Oriented Architecture (SOA) principles that  Is the foundation for future application development at VBA  Incorporates legacy data and applications, e.g. VETSNET |
| 7-5 | WINDOWS BASED GUI(DELPHI) | CAPRI | Graphical User Interface | [See Delphi Components and Extensions in this Document](file:///C:\Users\Jan\Downloads\CAPRI_DVBA_27_192_SDD_FINAL.docx#ColumnTitle_115) |
| 7-6 | HeathleVet Web Services Client | As VistA is migrated to the new Health*e*Vet-VistA system, some legacy VistA applications will need synchronous access to Health*e*Vet application services and data. HWSC uses Caché’s Web Services Client to invoke web service methods on external servers and retrieve results. It provides helper methods and classes to improve the use of the Web Service Client in a Health*e*Vet-VistA environment | Web Server Manager  Web Service Manager  Lookup Key Manager | HWSC acts as an adjunct to the web services client functionality provided in Caché, by:  -Leveraging Caché's platform-provided Web services client capabilities.  -Adding a file and UI to manage the set of external web server endpoints (IP, port, etc.)  -Adding a file and UI to register and manage the set of external web services.  -Providing runtime API to invoke a specific web service on a specific web server.  -Providing a runtime API to facilitate error processing in a VistA environment.  -Providing a deployment API to install/register a web service proxy from a WSDL file.  -Providing a management UI including the ability to ‘ping’ (test) a given web service/server combination from VistA/M.   * Supporting both SOAP- and REST-style web services   Fostering consistent implementation of VistA/M web service consumers |
| 7-6  cont |  |  |  | -Providing a management UI including the ability to 'ping' (test) a given web service/server combination from VistA/M.  Supporting both SOAP- and REST-style web services  Fostering consistent implementation of VistA/M web service |
| 7-7 | VENDOR LOCATIONS |  |  |  |
| 7-8 | MULTIPLE VA ACCESS | Links to multiple VAMCs from a drop down list to bring up that sites access logon prompt. Access credentials granted by the local ISO. The CAPRI Remote site selection screen displays the user’s authorized VHA facilities When the user selects a CAPRI Remote site executable it provides the authorized remote sites. If shown, the vertical scrollbar is used to scroll through all authorized sites. The user selects a site and then either double-clicks the site’s name or clicks **OK** to access that site. CAPRI has been modified to include the city and state where each facility is located. In addition, the list may now be sorted by State. | CAPRI GUI FORM/ DROP DOWN LIST | . (These accesses are established when an account is initially created and/or the user request specific facilities along with the proper approvals after the account creation. This facility information is located in the CAPRI file 396.96.) |
| 7-9 | CAPRI REMOTE USERS | Most VBA users are CAPRI Remote users. Each CAPRI Remote user needs only one Access Code and one Verify Code to connect to authorized VA Medical Center (VAMC) sites. CAPRI Remote users outside of VBA will normally obtain an access code from the Office of Information (OI) support staff, not from the local field site | Remote Access | The server for those users should be set to **CLAIMS.MED.DNS** , port **9400** |
|  |  |  |  |  |
| 7-10 | FHIE/BHIE  FRAMEWORK | Federal Health Information Exchange (FHIE)/ Supports monthly, 1-time transfer of electronic health information from DoD to VA at the point of a Service member’s separation.  Bidirectional Health Information Exchange (BHIE) Enables real-time sharing of electronic health information between DoD and VA for shared patients | *FHIE One-way, enterprise exchange of text data*  BHIE *Two-way, BHIE sites and all of VA exchange of text data* | FHIE: Over 3 million unique patients  Over 45 million lab results  Over 7 million radiology reports  Over 45 million pharmacy records  Over 41 million standard ambulatory data records  Over 2 million separated Service members have presented to the VA  ~2,700 messages transmitted daily in VA patients treated by DoD under local sharing agreements |
| 7-11 | DEPARTMENT  OF  DEFENSE | The DoD and the VA have been working together, starting in 2002, to share health information between their two Electronic Health Record (EHR) systems. The VA’s EHR is Veterans Health Information Systems and Technology Architecture / Computerized Patient Record System (VistA CPRS) and the DoD’s is Armed Forces Health Longitudinal Technology Application (AHLTA), formerly CHCS II, US DoD military health system. Transferring information between VistA CPRS and AHLTA was a major undertaking, since both EHRs were created using different types of software applications.  Both the DoD and the VA have been committed to the task and worked together to develop advanced technology to ensure the secure transfer of electronic health information between the two EHR systems. This important work supports the health care provided to active duty service members and veterans | Remote Information transfer | When a capri patient is selected the list of facilities care was provided will include Station 200 if DOD Data is available. Station 200 queries the BHIE framework for needed data element for presentation to the connected VA in the selected report type. |
| 7-12 | VLER/DAS  JAVA AND DELPHI | This capability with the eHealth Exchange renders VistAWeb a key component of Virtual Lifetime Electronic Record (VLER) electronic health information exchange. Uses Object Oriented Programming to access Vista Pateint data. | Web Service | VLER/DAS web service requires two-way SSL (also known as mutual authentication). CAPRI has been modified to relay VLER DAS web service requests through the CAPRI Proxy service using one-way SSL.  libeay32.dll VLER/DAS SSL dynamically linked library  - ssleay32.dll VLER/DAS SSL dynamically linked library |

Table 7-2 – Delphi Components and Extensions

| **File extension** | **Type** | **Delphi Version** | **Description** |
| --- | --- | --- | --- |
| .bdsdeploy | ? |  | Deployment file |
| .bdsgroup | text/xml | Delphi BDS 2006 and earlier | BDS Project Group, combines several projects into a group. Only put under source control if you actually use that feature. |
| .bdsproj | text/xml | Delphi BDS 2006 and earlier | Borland Developer Studio **Project File**. Successor of the .dof file holding compiler options etc. Also used for opening a project. |
| .bpg | text | Delphi 7 and older | Borland Project Group, combines several projects into a group. Only put under source control if you actually use that feature. |
| .bpl | Binary | Delphi 4 and newer | Borland **Package Library**. DLL including VCL components available for use in the Delphi environment at design time or by applications at run time. |
| .cbk | Other |  | **Temporary Backup** File. |
| .cfg | text | Delphi 7 and older | **Project Configuration**File. The Delphi IDE (up to Version 7) automatically creates this file from the settings in the .DOF file. It is only used by the command line compiler. See [command line compilation](http://delphi.wikia.com/wiki/Compile_from_Commandline) . |
| .config |  |  | Config files contain **project option information** and **build logic**. Each project has a .config file. |
| .d | text |  | Output of the project unit dependencies. This is generated by checking the "Output unit dependency information" compile option or passing "-depends" dcc32.exe. |
| .dcl | binary |  | Delphi Component Library |
| .dcp | Binary |  | Delphi **Component Package**. Contains symbol information for code compiled into a package. Used if building with runtime packages. |
| .dcpil | Binary |  | Delphi Compiler **Symbolic Information**. The purpose of a .DCPIL file is to provide symbolic information to the compiler about certain language constructs that can't be reconstituted from CLR metadata alone. This is a traditional Delphi technique for handling and caching compiler symbols. |
| .dcr | Resource |  | Delphi **Component Resource** File. Contains resource information for components, such as the icon to display in the palette. |
| .dcu | Binary | Win32 | Delphi **Compiled Unit** File. |
| .dcuil | Binary | .NET | Delphi **Compiled Unit** File |
| .ddp | Other | Delphi 7 and older | Delphi **Diagram Portfolio** File. Used by the diagram editor. |
| .dfm | binary or text |  | Delphi **Win32 Form** File. A file listing the various objects and their property values contained within a form, including such things as form control coordinates. Right click on the form and select View as Text from the pop-up menu to view this file. Manual alterations to this file may prevent the IDE from being able to load the form. Use the Environment Options dialog to specify which format to use for newly created forms. The text format is recommended as it better supports source control. |
| .dof | text | Delphi 7 and older | Delphi **Project Options** File. Text file containing project options (such as compiler and linker settings, version info, search path and output directories). |
| .dpc | ? |  | Delphi **Package Collection** |
| .dpk | text |  | Delphi **Package Project** |
| .dpkw | text |  | Delphi **Package Project** for Windows (Never seen in the wild) |
| .dpl | Binary | Delphi 3 and older | Delphi Package Library. Superseded by .bpl in later Delphi versions. |
| .dpr | text | All Delphi versions | Delphi **Project File**. The main file for any project. Contains the source code of the main file for a Delphi project. It is the primary entry point for the executable and refers to the other source files in the project. Do not delete. |
| .dproj | text/xml | Delphi 2007 and newer | Delphi **Project File for the native compiler**. Contains the current settings for project options, such as compiler and linker settings, directories, conditional directives, and command-line parameters. Set these options using Project > Options. Replaces bdsproj file. |
| .drc | text |  | Resource String File. Contains resource string information. Can be deleted; will be regenerated from source code by compiler. |
| .dres |  |  | Delphi **compiled resource file**. |
| .dsk | text |  | Delphi **Desktop File**. Contains information about the IDE desktop settings, including which files are open and the position of windows. |
| .dsm | Binary | Delphi 6? | **Browser Symbol** information from the last successful compile. Enviroment -> Preferences -> Desktop contents. |
| .dst |  |  | File used to save the desktop speed setting as set in the IDE toolbar desktop combo box. |
| .groupproj | text/xml | Delphi 2007 and newer | **Project Group** File. Replaces .bdsgroup file. |
| .identcache | Binary |  | **Temporary Cache** File created by refactoring engine to improve performance. |
| .int | text |  | Interface part of Unit not shipped with source, typically in $(Delphi)\doc |
| .local | text/xml | Delphi 2005 and newer | User-specific **project options**. When using Starteam projects it will contain overrides to settings from the .bdsproj file. |
| .map | text |  | The optional **error map** file will contain a list of error addresses and source line numbers, grouped together for each source file in the application. To create a detailed .map file, ensure the {D+} compiler directive is set, and the "Detailed Map File" option selected in the Project >Options>Linker page of the IDE. |
| .mts | QA Metrics set file | Rad Studio 2010 enterprise or newer | The stored settings for QA metric |
| .nfm | Resource | Delphi 8.0 or newer | Delphi **.NET Form** File. |
| .pas | Source |  | Delphi **Unit Source** File. |
| .proj |  |  | Codegear RAD Studio project. |
| .res | Binary |  | *Binary*. **Resource File**. Binary file associated with a project containing resource definitions, such as strings, icons, images etc.This file will be recreated by the IDE when loading a project but the icon will get lost in that process. Therefore you should either keep it in source control or find another way to create it. |
| .resources | Binary |  | A binary **resource file** that can be embedded in a runtime executable. |
| .rsm | Binary |  | Remote Debugging Symbols file. |
| .tlb | Binary |  | **Type library**. |
| .todo |  |  | The **project to-do list**. |
| .tvsconfig | text/xml |  | Modeling configuration file. |
| txvpck, txvcls |  |  | Information for model diagram. |
| .vlb | text/ini |  | Visual Live Bindings |

## Interface Detailed Design

Figure 7 - 2 – Interface Detailed Design



**Table 7-2 – Interface Detailed Design Descriptions**

| ID | Interface Name | Description | Interface type |
| --- | --- | --- | --- |
| 7-13 | View Procedures | When the user selects the clinical documents tab another complete set of tabs opens along the bottom of the screen Selecting the **Clinical Documents** tab yields access to View Procedures(Holter Monitor, Echocardiogram, Electrocardiogram, Thallium Stress Test, etc.) | CAPRI GUI Clinical Documents Tab |
| 7-14 | Load new patients into VistA system | VBA users should only establish a new patient within the VistA system if it is necessary to request a C&P examination for a veteran who is not a current patient within the medical facility’s database. If the patient NAME and SSN search using CAPRI’s Patient Selector function yields no results, i.e., “Match not found,” then the user must establish the veteran as a new patient | CAPRI GUI Patient Selector Form |
| 7-15 | View lab reports | When the user selects the clinical documents tab another complete set of tabs opens along the bottom of the screen Selecting the **Clinical Documents** tab yields access to Laboratory Findings | CAPRI GUI Reports selection Form/Tab |
| 7-16 | View progress notes.. | Notes Tab  This tab allows the user to view or print the veteran’s progress notes . The left side of the screen shows the appointment date, type of progress note, and the author of the note. If the user places the mouse pointer over a progress note, the full title is displayed. When a note is selected, it appears in the window on the right. | CAPRI GUI  Notes Tab |
| 7-17 | View imaging | When the user selects the clinical documents tab another complete set of tabs opens along the bottom of the screen Selecting the **Clinical Documents** tab yields access to Imaging (X-rays, CT, MRI, etc. | CAPRI GUI Clinical Documents Tab |
| 7-18 | View discharge summaries | When the user selects the clinical documents tab another complete set of tabs opens along the bottom of the screen Selecting the **Clinical Documents** tab yields access to Discharge Summaries |  |
| 7-19 | View cumulative vitals | This creates a report that matches the data generated by the AMIE option of the same name. It is a cumulative report containing all admissions for a given date range. The report is designed primarily as an auditing tool for the RO. Information which may be provided for each patient includes claim number, claim folder location, SSN, admission date, admitting diagnosis, discharge date, bed service, whether the patient is receiving A&A or pension, and eligibility data. Depending on the date range selected, this report can be quite lengthy.  **Step 1** – The user enters a date range using Start Date and Stop Date.  **Step 2** – A specific regional office is selected.  **Step 3** – The user clicks **OK** | CAPRI GUI Reports selection Form/Tab |
| 7-20 | VA  Regional Office reports | AMIS 290This option electronically produces the AMIS C&P report that was manually produced by the regional office. It is a general system of computer programs used to process management reports. The AMIS 290 report covers compensation and pension examination request activity. The regional office AMIS 290 calculates the data based only on that specific regional office’s requests, or when the regional office is left blank, it calculates the data for all regional offices | CAPRI GUI REPORTS Form |
| 7-21 | Automatic Mailman bulletins to mail groups AMIE | AMIS 290, Processing time for an insufficient request includes the processing time of the original request. In addition to a hard copy being produced, this option allows the user to send a MailMan message either locally or via network mail. The mail bulletin contains the same information that appears on the report. Vocational Rehabilitation Tab- Vocational Rehabilitation and Employment (VR&E) employees require the ability to refer a veteran to a medical center for the purpose of specific types of exams or opinions in order to determine VR&E benefits.  This feature enables users to create an electronic request for medical services, track those requests and print reports. A consult is created outside the CAPRI application. The request can be linked to the consult by a VHA Coordinator. Notifications are sent to both VR&E Counselor (via Outlook) and VHA Coordinator (via MAILMAN) when the 28-8861 form status is updated. A nightly background job monitors consults that are linked to the 28-8861 file. Specific Events Trigger Mailman Messages; Upon specific events, MAILMAN messages are sent to **VHA Coordinators** who **must** be a member of the | C&P Exams Reports, AMIS 290; Vocational Rehabilitation Tab;Specific Events |
| 7-21 continued | Automatic Mailman bulletins to mail groups AMIE (Continued) | MAILMAN mail group: DVBA VR VOCREHAB PERSONNEL and the VR&E Counselor(s) receive email notifications via Outlook These notifications are sent for the following status changes to the Medical Services Request:  -New Status: This message reads: Chapter 31 Referral for Medical Services New. \*\*This notification is only sent to the VHA Coordinator via MAILMAN  -Pending Status: This message reads: Chapter 31 Referral for Medical Services Pending. \*\*In CAPRI, once the VHA Coordinator/or designated person links the consult request to the original Chapter 31, VA Form 28-8861 request, the status of the consult in CAPRI is then changed to Pending. Both the VHA Coordinator and VR&E Counselor receive notification messages  -Completed Status: This message reads: Chapter 31 Referral for Medical Services Completed \*\* Both the VHA Coordinator and VR&E Counselor receive notification messages  -Cancelled Status: This message reads: Chapter 31 Referral for Medical Services Cancelled\*\* Both the VHA Coordinator and VR&E Counselor receive notification messages |  |
| 7-22 | View patient demographics. | Individual Patient Demographics can edited after selecting the patient from the CAPRI Patient Selector Form | CAPRI GUI CAPRI Patient Selector Form |
| 7-23 | Report patient address changes to VHA | Address Verification Form; CAPRI repeatedly prompts the user to check the latest address of record in the VistA system. If the veteran has a new address, the user clicks the **Edit Address Now** button. If the user is not sure if the new address is more current than the one shown by CAPRI, then the Permanent Address is not updated and instead this fact is noted and the new address is included in the general comments area of the exam request.  If the **Permanent Address:** shown by CAPRI is current, the user clicks **OK**. If the user enters a change to the **Permanent Address**: information, it does not update the address information directly in the VistA Registration database; instead this information is appended as text to the examination request, where it can be further edited if necessary. | CAPRI GUI Form  Address Validation Form |
| 7-24 | Add/Edit C&P exam request | CAPRI prompts the user to check the latest address of record in the VistA system. If the veteran has a new address, the user clicks the **Edit Address Now** button. If the user is not sure if the new address is more current than the one shown by CAPRI, then the Permanent Address is not updated and instead this fact is noted and the new address is included in the general comments area of the exam request.  If the **Permanent Address:** shown by CAPRI is current, the user clicks **OK**. If the user enters a change to the **Permanent Address**: information, it does not update the address information directly in the VistA Registration database; instead this information is appended as text to the examination request, where it can be further edited if necessary. | CAPRI GUI Add/Edit C&P Exam request form |
| 7-25 | Multiple templates can be merged into a single exam | CAPRI prompts the user to check the latest address of record in the VistA system. If the veteran has a new address, the user clicks the **Edit Address Now** button. If the user is not sure if the new address is more current than the one shown by CAPRI, then the Permanent Address is not updated and instead this fact is noted and the new address is included in the general comments area of the exam request.  If the **Permanent Address:** shown by CAPRI is current, the user clicks **OK**. If the user enters a change to the **Permanent Address**: information, it does not update the address information directly in the VistA Registration database; instead this information is appended as text to the examination request, where it can be further edited if necessary. | CAPRI GUI Browse Templates Form |
| 7-26 | View FHIE/DoD data, if available | CAPRI Remote users have access to the DoD Tab. From this tab, DoD records are available for certain veterans beginning approximately six weeks after discharge. These records are available through the Federal Health Information Exchange (FHIE). The user knows if DoD records are available when they access the Patient Selector screen and the notation **DOD data is available** indicates that the user can access these records after selecting the patient. The patient demographic data is not visible in this specific example because it is a sensitive level record. | CAPRI GUI Tabbed Form DoD Records Tab |
| 7-27 | Patient Records Navigation | Patent Selector Screen Local and Remote Sites give the ability to Navigate patient records to get a more complete understanding of care | CAPRI GUI Patient Selector Form |
| 7-28 | C&P Examination Functionality | C&P Exam Requests  The C&P Exam tab includes functions such as:  -Requesting C&P examinations  -Viewing/editing requests  -Canceling requests  -Adding an exam to a pending request  -Creating status inquiry reports  -Viewing completed requests  -Tracking the progress of the request for claims management purposes  -Requesting an Insufficient Exam-  -Printing results for individual patients  Pending requests are shown with a date only in the left column. Completed requests have a completion date in the right column. When the user first accesses this screen, none of the examinations are selected, and only the **Add a New Request** button is enabled. After the user selects an examination, the **Re-Print Final C&P Results**, **Status Inquiry**, and **View/Edit Selected Request** buttons are enabled. | CAPRI GUI C&P Exam Request Form |
| 7-29 | All Standard AMIE worksheets are available in template form | Selecting **AMIE Worksheets** takes the user to a C&P website that provides Index to Disability Examination Worksheets. Selecting a worksheet displays instructions for using the worksheets | CAPRI GUI Tools Tab/ Web Services ; |
| 7-30 | View active medication | When the user selects the clinical documents tab another complete set of tabs opens along the bottom of the screen Selecting the **Clinical Documents** tab yields access to View Active Medications | CAPR GUI Reports Selection Form/Tab |
| 7-31 | View Appointments Lists | Patient Profile Medical Administration Service (MAS) (Full)  The following screen is the default – it specifies all dates, appointments, enrollments, team information, edits, dispositions, and the means test. The user can change the generated report to exclude particular types of information by selecting **No** for that particular type | CAPR GUI Reports Selection Form/Tab |
| 7-32 | View health summaries | Health Summaries are customized reports comprising VistA components specified by end users. Most of these summaries were developed by the VHA facilities. Regional Offices can create special summaries that appear on the menu. The Veterans Administration Regional Office (VARO) St. Petersburg, in cooperation with the Veterans Integrated Service Network (VISN) 8, developed a Health Summary called VARO Rating, which contains components specified by RVSRs to facilitate their work process. To develop a custom Health Summary, the user must contact the VHA facility’s IRM. The VARO Rating Health Summary in VISN 8 facilities include the following VistA components: demographics, imaging impressions, past and future clinic appointments, admissions and discharges, discharge summaries, progress notes, surgery reports, and medications. In addition to the reports on the menu, the user can create one-time Ad Hoc reports for use with a particular case | CAPR GUI Health Summaries Tab |
| 7-33 | Request VAF 7131 information | 7131 requests can be made for reports including:  -Patient records which may be retired after a long period of facility inactivity  -Patient records which only exist on paper  -VAF 21-2680 Aid and Attendance examinations that have been completed by the veteran’s health care provide;Competency reports;Asset information;21-day certificates;Records based upon hospital admissions such as discharge notices and discharge summaries | CAPRI GUI **7131 Request** tab |
| 7-34 | AMIS 290 report | The AMIS 290 report is no longer available in CAPRI. Selecting this report directs the user to a web location where AMIS 290 data is available. http://DNS demo.DNS /dmareports.asp | CAPR GUI Reports Selection Form/Tab |
| 7-35 | Export Excel delimited reports | Reports options The data can be downloaded from these reports in text delimited or comma separated value file formats so that further analysis can be performed on the data such as searching and sorting; where these capabilities are provided by external applications capable of accepting the data (e.g.: Microsoft Access® or Microsoft Excel®). | CAPR GUI Reports Selection Form/Tab |
| 7-36 | Automatic Sending of completed exam requests | When an examiner completes an exam from the signature validation screen it will automatically be transmitted to Virtual VA. The examiner will need to acknowledge the **Virtual VA Transmission Status window** as shown below by clicking the “Close” button |  |
| 7-37 | View consult requests and results | Notification Messages: CAPRI, once the VHA Coordinator/or designated person links the consult request to the original Chapter 31, VA Form 28-8861 request, the status of the consult in CAPRI is then changed to **Pending**. Both the VHA Coordinator and VR&E Counselor receive notification messages |  |
| 7-38 | Export PDF | Documents handled by CAPRI can be in Portable Document Format |  |
| 7-39 | Demographics | View Registration Data Report  This report provides full demographic data, including military information, for the selected patient.  Health Summaries Tab; Health Summaries are customized reports comprising VistA components specified by end users. Most of these summaries were developed by the VHA facilities. Regional Offices can create special summaries that appear on the menu. The Veterans Administration Regional Office (VARO) St. Petersburg, in cooperation with the Veterans Integrated Service Network (VISN) 8, developed a Health Summary called VARO Rating, which contains components specified by RVSRs to facilitate their work process. To develop a custom Health Summary, the user must contact the VHA facility’s IRM. The VARO Rating Health Summary in VISN 8 facilities include the following VistA components: demographics, imaging impressions, past and future clinic appointments, admissions and discharges, discharge summaries, progress notes, surgery reports, and medications. In addition to the reports on the menu, the user can create one-time Ad Hoc reports for use with a particular case. | CAPR GUI Reports Selection Form/Tab;  Health Summaries Tab; |
| 7-40 | Individual and cumulative pending exam tracking | C&P Examination Functionality  CAPRI automatically checks pending Compensation and Pension Worksheet Module (CPWM) Template statuses. Pending templates in the user’s queue are displayed on the alert screen. The C&P Alert screen displays alerts according to template status. Alerts for template statuses are: **draft**, **awaiting signature**, **sent back from reviewer**, **requiring review**, **CPRS documents to cosign**, and **cosigned documents ready to transfer to AMIE**. | CAPRI Alerts |
| 7-41 | Create an insufficient exam request | **Add An Insufficient Exam Request** if this examination was completed but is insufficient This option is used if a completed examination is not sufficient, and the RO has followed local procedures to attempt to make the examination sufficient for rating purposes, The user verifies that the correct request was selected, and then clicks the **Add An Insufficient Exam Request** button. – The **Add New C&P Exam** screen opens . The user completes this screen as directed in Section 2.2.3, **Add a New Request**. The only difference is that the list of available examinations is limited to those completed in the previous, insufficient examination. These choices only appear after the user enters the **Routing Location** | CAPRI GUI View C&P Exam “Edit” form |
| 7-42 | Export Plain Text Reports | Reports options The data can be downloaded from these reports in text delimited | CAPR GUI Reports Selection Form |
| 7-43 | Ability to save template work in progress and finish later | A details bar is available on each free text area of the template. The details bar is designed to provide users with a larger data entry area. The details bar becomes visible when the mouse is over the text box. After the user clicks the details bar, a larger window is displayed. The user can then type the information desired and save and close the details box. When the details box is closed, the information is displayed in the original free text area | Save through RPC from GUI to VISTA |
| 7-44 | Ability for site to review exams before releasing  it to VBA | Some sites require the exam to be sent to a reviewer when it is complete. One and only one of the following Security Keys must be assigned for anyone who is creating new C&P exam templates with CPWM.  1. DVBAB CPWM DISALLOW REVIEW – User does not need documents reviewed prior to release.  2. DVBAB CPWM OPTIONAL REVIEW – User can choose to send some documents for review and not other documents.  3. DVBAB CPWM REQUIRE REVIEW – User must have all documents reviewed by a reviewer prior to upload.  **OPTIONAL** **Security** **Key** –If the site chooses to use the review process, users designated as reviewers must be assigned the **DVBAB CPWM REVIEWER** Security Key | VISTA Option and Key assignment |

# Human-Machine Interface

CAPRI provides VBA employees with a standardized, user-friendly method to rapidly access veterans' electronic medical records throughout the Department of Veterans Affairs (VA). CAPRI delivers leading edge "point and click" technology to the users' desktops. In addition, the learning curve for CAPRI is significantly less than that for character-based systems.

Figure 8 - 1 - Human Machine Interface

| **HUMAN** | | **HARDWARE** | | **SOFTWARE** | | |
| --- | --- | --- | --- | --- | --- | --- |
| Brain | Eyes  Ears  Hands | Peripheral Hardware | Integrated Circuits | Kernel space | User Space |  |
| Display Monitor,  Speakers | 🡨 Output | Video Subsystem.  Audio Subsystem | VA Network,  VistA, CAPRI Application | 🡪  Input-Output Loop  🡨 |
| Keyboard, Mouse | Input 🡪 | Input Subsystem | ^ |  |

## Interface Design Rules

CAPRI GUI follows standard widows application design conventions.

The GUI utilizes standard, 508 accessible, screen controls in all it’s forms and dialogs.

Most commonly used controls are as follows

* Tabbed Panels
* Labels
* Edit boxes
* Pulldown controls
* Buttons
* Menus (common elements like “File”, “Help”, “Help -> About”

## Inputs

All user input into the CAPRI application is via the GUI. Access to various entry screens or menu options that allow opening of some screens may be restricted by VistA Security Keys as managed by Local IT/IRM.

### Logging onto CAPRI

The information is found in the latest version of the CAPRI GUI User Manual, see section “Logging On” at the following URL: [http://www.DNS /vdl/application.asp?appid=133](http://www.DNS   /vdl/application.asp?appid=133)

### VistA Terminal

The “Attachmate Reflections Secure Shell” application replaces the telnet window. CAPRI Remote users can launch a VistA Terminal session by selecting the VistA button to log into the local VistA system site they are assigned to. A dialog box is displayed when the VistA button is clicked that provides the user the ability to choose between connecting using the secure shell application or telnet. The default is set to secure shell application. Telnet is not an authorized method of connection.

Note: Local CAPRI users will not have access to the VistA Terminal from CAPRI and the VistA button will not be available.

## Using the Software

### Patient Selector Screen or Patient Entry / Selection

***Security Note:*** *The CAPRI application does not allow users to view their own personal patient records. If a user attempts to do so, CAPRI prevents access and alerts the Security Administrator at the VHA facility who takes established security violation actions. In addition, when selecting a patient who is a VA employee, CAPRI allows the user to do so only after the user agrees to Privacy Act Terms via a dialog box. An alert is sent to the Security Administrator who then inquires about the user’ s business reasons for accessing those records****.***

### Selecting a Patient / Veteran

The Patient Selector screen allows the user to search the VistA database for patients who have records. Patients can be located using a Social Security Number (SSN), the last name initial plus the last four digits of the SSN, or by typing in the veteran’s last name and first name. Current users of AMIE II will recognize that these are the same search methods for that application.

If the user is looking for existing VHA medical records and the patient selection search yields no results, then there are no existing VHA medical records for that patient at the facility that was accessed. Verify that the user typed in the SSN correctly and that the user is logged into the correct VHA facility.

***Note 1: To avoid displaying sensitive information regarding our patients and staff, the examples in this manual contain contrived names instead of real names. Patients and staff are referred to as CPRIPATIENT, ONE; PROVIDER, ONE; or USER, ONE. Likewise, real SSNs, real addresses, and other personal identifiers are not used.***

**Step 1** – To begin the search, the user clicks **File/Select Patient** and enters the veteran’s information (Figure 8 - 3). For example, if the user wants to view the records of CPRIPATIENT, TWO (SSN: 666080315), then the following would be the valid search methods:

* SSN (preferred method): the user enters 666080315 and clicks Select
* Last name initial and last four of SSN: the user enters C0315 and clicks Select
* Name: the user enters CPRIPATIENT, TWO and clicks Select

***Note 2: There is NO spaces between the comma and the first name.***

**Step 2** – CAPRI displays a list of possible matches (). The user can click once on the patient’s name to display more information to help verify that the user has selected the correct veteran. This additional information includes the patient’s full name, SSN, claim number, gender, age, and date of birth (DOB). If there is only one match, this information is displayed automatically. After choosing the correct veteran, the user clicks the **Select** button.

**Step 3** – After the user clicks **Select**, CAPRI builds all of the background information on the selected veteran and automatically opens the C&P Exam tab. From this starting point the user can request a C&P exam for the selected veteran or select any of the other tabs to navigate existing patient records.

Figure 8 - 3 – Patient Selector

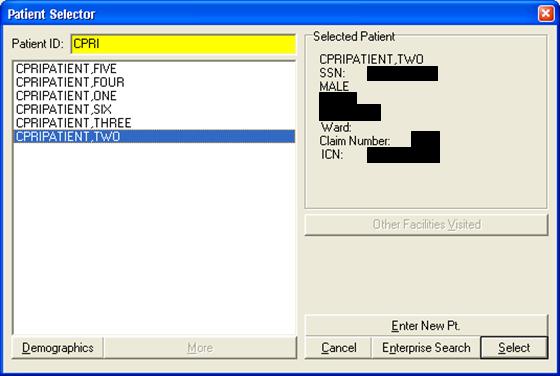
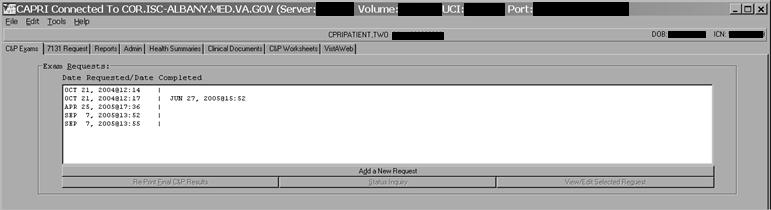


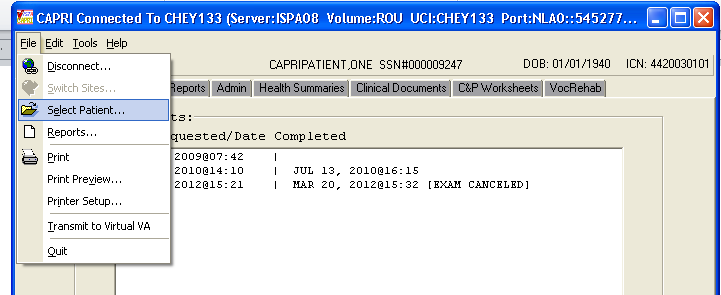
Figure 8 - 4 is the result screen after selecting a patient.

Figure 8 - 4 – Screen After Selecting a Patient



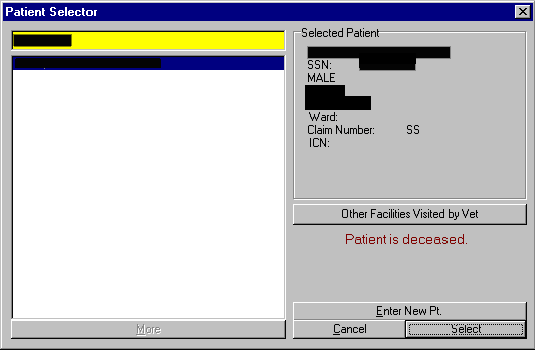
**Step 4** – To select another patient, the user selects **File/Select Patient** (Figure 8 - 5) and returns to **Step 1** of this section.

Figure 8 - 5 – CAPRI File Menu



***Note: CAPRI notifies the user if the patient is deceased and allows the user to Continue or Cancel. When the user searches using the Patient Selector screen, a message indicates if the patient is deceased (***Figure 8 - 6***).***

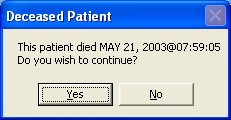
Figure 8 - 6 – Patient is deceased notification



If the user clicks **Select**, the following dialog box appears (Figure 8 - 7) with the patient’s date and time of death.

The user selects **Yes** to continue or **No** to cancel.

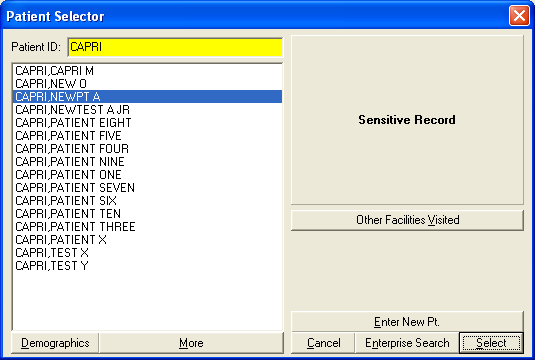
Figure 8 - 7 – Deceased Patient Dialog Box



### Sensitive Records

A user’s own record and the records of the user’s colleagues are designated as sensitive records. The user may not access them without justification. If a user clicks the name of patient whose record is sensitive, the words **Sensitive Record** (Figure 8 - 8) are displayed in place of the patient’s personal information.

Figure 8 - 8 – Sensitive Record



If the user attempts to open this record, a warning message is displayed (Figure 8 - 9).

If the user does not have justification for viewing this record, then the user should select **No** on this message box.

Figure 8 - 9 – Sensitive Record Dialog Box

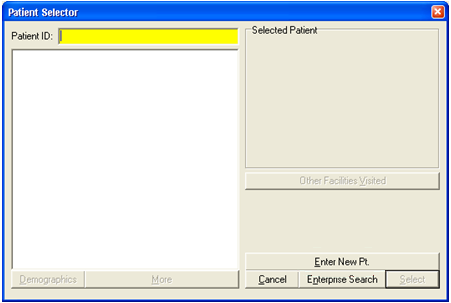


### Entering a New Patient / Veteran

***Note: VBA users should only establish a new patient within the VistA system if it is necessary to request a C&P examination for a veteran who is not a current patient within the medical facility’s database. If the patient NAME and SSN search using CAPRI’s Patient Selector function yields no results, i.e., “Match not found,” then the user must establish the veteran as a new patient.***

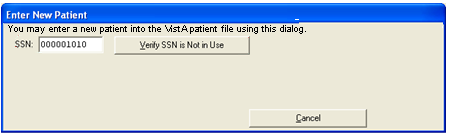
Step 1 – The user selects the **Enter New Pt.** button at the bottom of the Patient Selector screen (Figure 8 - 10).

Figure 8 - 10 – Enter New Patient



**Steps 2** – User enters the veteran’s SSN in the space provided, and then clicks **Verify SSN is Not in Use** button (Figure 8 - 11).

Figure 8 - 11 – Verify SSN Not in Use



**Step 3** – If the SSN is not associated with an established patient record, CAPRI opens the **Enter New Patient** template (Figure 8 - 12).

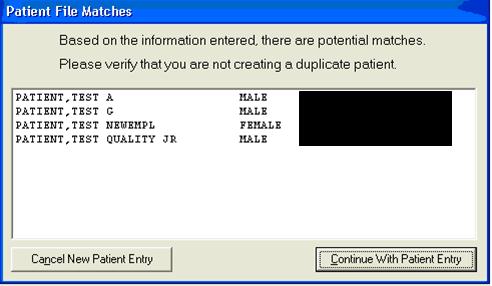
When the user selects the **Click to Continue** button, a query is run to find any existing patient with the following information similar to the new patient:

* First two letters of the First and Last name are the same
* Same last four of the SSN
* Same Year and Month of the Birthdate
* Same Year and Day of the Birthdate
* Figure 8 - 12 – Enter New Patient



The query results are displayed so that the user can manually verify that the new patient does not exist in the system. The **name, gender, DOB**, and **SSN** of the potential matches are displayed, as shown in (Figure 8 - 13).

Figure 8 - 13 – Patient File Matches



## Outputs

The following are outputs from the CAPRI Software:

* C&P Exam Requests
* 7131 Request
* Reports (Patient Specific, Non Patient Specific, Consolidated Remote Reports)

### C&P Exam Requests

The C&P Exam tab ( ) includes functions such as:

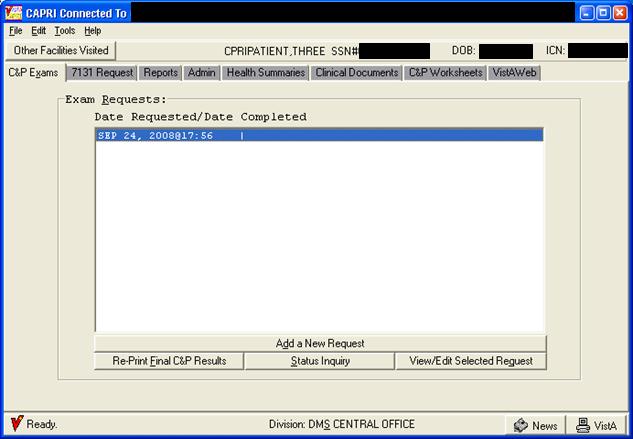
* Requesting C&P examinations
* Viewing/editing requests

#### View/Edit Selected Request

**Step 1** –The user logs into CAPRI, looks up the veteran using the **Patient Selector**, and moves to the **C&P Exams** tab (Figure 8 - 14). If there are no examination requests, the window is blank. If the veteran already has examination requests on file, the window shows the dates requested.

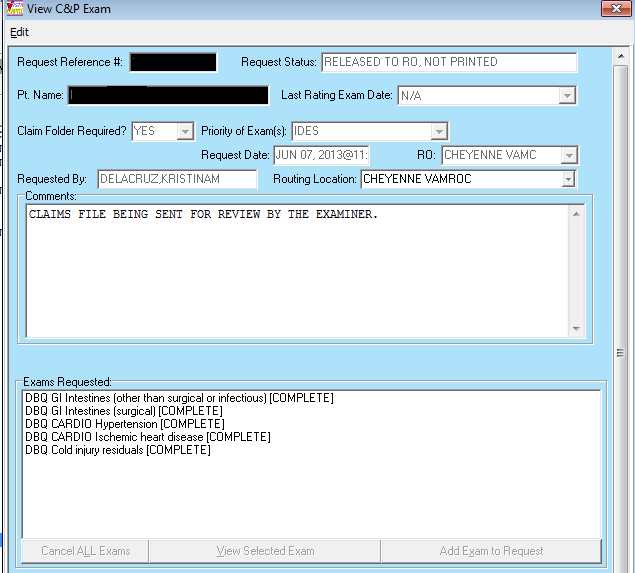
**Step 2** –The user selects the examination request date and clicks the **View/Edit Selected Request** button.

Figure 8 - 14 – View/Edit Selected Request



**Step 3** – The **View C&P** screen opens, showing all of the entries from the original request, as well as the examination status (Figure 8 - 15). The user can scroll down to see additional information from the original request. The exam in this request cannot be edited because it is already complete, but the original request may be viewed.

Figure 8 - 15 – View C&P Exam



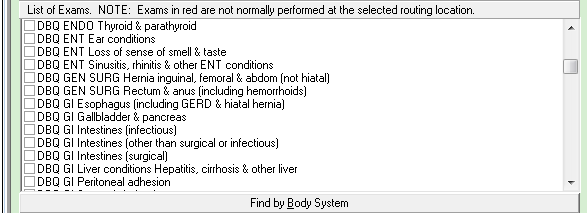
**Step 4** – If the user wants to edit a request that is still pending and has not yet been scheduled, it can be done on the **View C&P Exam** screen shown in Figure 8-13. Otherwise, skip to **Step 8**. The user can edit **Claim Folder Required?,** **Priority of Exam(s)**, **Routing Location**, **Comments**, and **Other Disabilities** by typing directly into those fields. To add another examination, the user clicks the **Add Exam to Request** button shown in (Figure 8 - 16).

Figure 8 - 16 – Add Exam To Request

CAPRI Screen Excerpt - Cancel / View / Add button

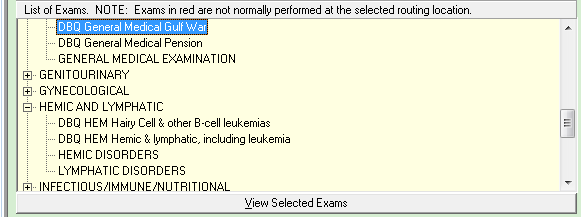
**Step 5** – If the **Add Exam to Request** button is clicked, the exam listing field opens allowing the user to make a selection (Figure 8 - 17). The user can scroll down the list, or use the **Find by Body System** button. In the **Find by Body System** view (Figure 8-16), the user scrolls down the body systems until the correct one is found, then single-click the “**+**” box in front of the body system name, or the user can double-click the body system name to open a list of all pertinent examinations. The user double-clicks the desired examination to add it. The view reverts to the scroll list view, and that examination is checked.

Figure 8 - 17 – List of Exams



**Step 6** – In the scroll list view (Figure 8 - 18), the user clicks the checkbox next to any of the desired examination.

Figure 8 - 18 – List of Exams by Body System



**Step 7** –The user selects **Add Selected Exams** or **Cancel Add Exams** as appropriate (Figure 8 - 19).

**Step 8** –The user can select **Close Window** to close this screen or **Add An Insufficient Exam Request** if this examination was completed but is insufficient (Figure 8 - 20).

### 7131 Request

The 7131 Request tabincludes functions such as:

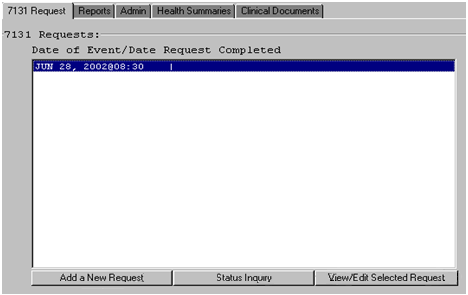
* Adding new requests for 21-Day Certificates, Notices of Discharge, etc…
* Status inquiries and reports
* Viewing and editing pending requests

*Note: 7131 requests are reserved for information that cannot be obtained directly through CAPRI, such as older records and retired records that may not exist in the electronic database or other records in VHA’s CPRS that are unavailable in CAPRI. For example, scanned records in CPRS may not be available in CAPRI.*

Pending requests are shown in the left column and completed requests are shown in the right column of the screen. In the following example (Figure 8‑18), the request is still pending for an Admission Report, and there are no completed requests. When the tab is first opened, all pending and completed requests are shown, but none is selected. The **Status Inquiry** and **View/Edit Selected Request** buttons are not available until a request is selected.

Figure 8‑21 shows what an open 7131 request for an outpatient treatment record looks like. The **Date of Event/Date Request Completed** displays **Out Patient (O/P) Activity** when the 7131 request is not for an **Admission Report**.

Figure 8 - 19 – 7131 Request



#### Add a New 7131 Request

7131 requests can be made for reports including:

* Patient records which may be retired after a long period of facility inactivity
* Patient records which only exist on paper
* VAF 21-2680 Aid and Attendance examinations that have been completed by the veteran’s health care provider
* Competency reports
* Asset information
* 21-day certificates
* Records based upon hospital admissions such as discharge notices and discharge summaries

To add a new 7131 request, do the following:

**Step 1** – The user logs into CAPRI, selects a patient, and clicks the **7131 Request** tab.

**Step 2** – The user clicks the **Add a New Request** button (Figure 8 - 19).

**Step 3** –The user selects either **Outpatient/Activity** or **Admission** Figure 8 - 20) depending on which one most closely relates to the request. If no admission is listed under the **Select Admission Date** (Figure 8 - 21) then an electronic 7131 Request cannot be submitted.

*Note: Notice of Discharge, Hospital Summary, Certificate (21-Day), and Admission Report are NOT available if the user selects Outpatient/Activity. These options are only available if Admission is selected.*

Figure 8 - 20 – New 7131 Outpatient Request

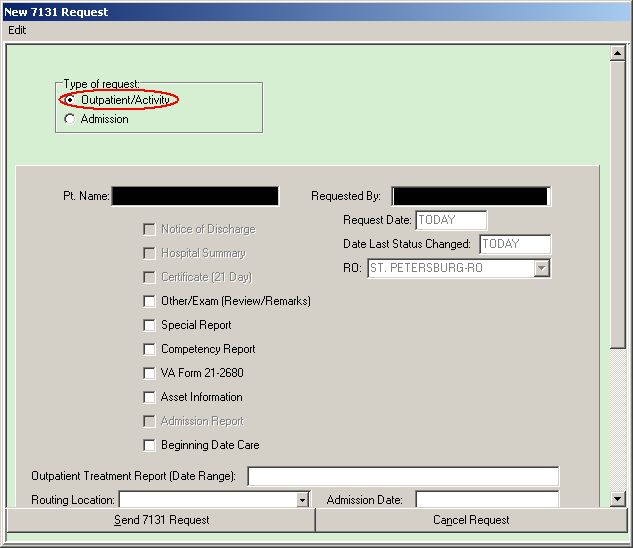
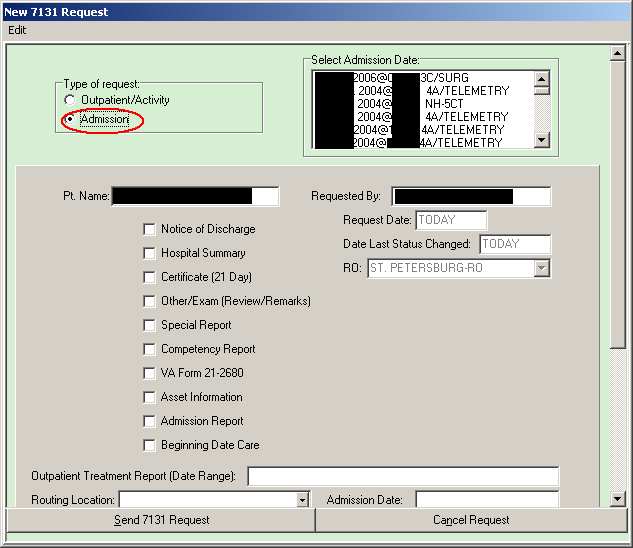


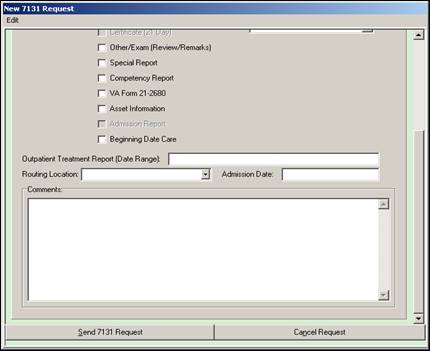
Figure 8 - 21 – New 7131 Admission Request



**Step 4** –The user checks the items to request i.e. **Pt. Name**, **Requested By**, **Request Date**, **Date Last Status Changed, RO**, and **Admission Date**.

**Step 5** –The user enters detail information concerning the request in the **Comments** area (Figure 8 - 22). A Release of Information Clerk at VHA must search old records for the user’s request and appreciates any help the user can give by being specific about the request.

**Step 6** –The user clicks **Send 7131 Request** and the request is transmitted. Or, the user can click **Cancel Request** (Figure 8 - 22). CAPRI opens a dialog box asking the user to confirm the choice.

**Figure 8 - 22 – Send 7131 Request**

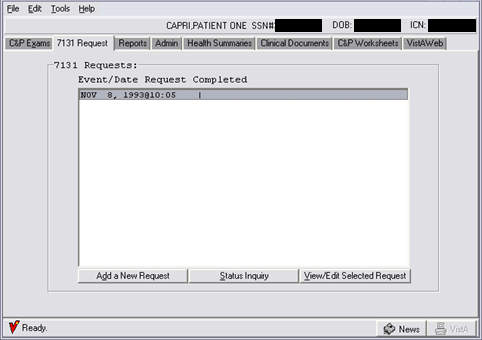
**Step 7** – To inquire about the status of a 7131 request or to generate a report for proof of a request to be placed in the claims file, the user selects the **Status Inquiry** option shown in the next section in 21.

#### 7131 Request Status Inquiry

Step 1 – The user logs into CAPRI, selects a patient, and clicks the 7131 Request tab. The 7131 Requests: screen displays any pending or completed 7131 requests (Figure 8 - 23).

Step 2 – The user selects the 7131 request which they would like to run the inquiry and clicks the Status Inquiry button.

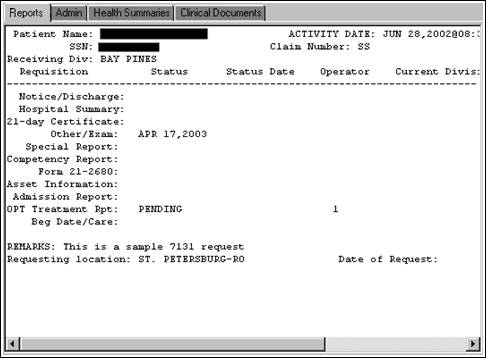
Figure 8 - 23 – 7131 Request Status Inquiry



Step 3 – The following report opens under the Reports tab (Figure 8 - 24). To check the status of another 7131 request, the user must go back to the 7131 Request tab and start from Step 2 of this section.

Step 4 – The user can print the report for the claim folder by choosing Print from File on the main menu.

Figure 8 - 24 – 7131 Request Report



### CAPRI Reports (Patient-Specific)

The **Reports** tab allows the user to make various patient inquiries, review registration and profile data, and request surgery reports for the selected veteran.

#### Patient Inquiry

This report provides demographic, eligibility, and treatment information about the selected patient.

#### Detailed Inpatient Inquiry

This report lists all of the patient’s inpatient admissions, with the date, time, and ward. The user can select an admission and click **OK** to get detailed information about that admission. This includes admission and discharge dates, transfers between wards, care providers, and diagnosis.

#### C&P Exam Detail

This report was moved to the **C&P Exams** tab. If attempting to use this report, the user is directed to that tab.

#### 7131 Detal

This report was moved to the **7131 Request** tab. If attempting to use this report, the user is directed to that tab.

#### Additional Treating Facilities

This report shows if the patient was treated at any VHA facilities other than the one currently connected to by the user.

#### View Registration Data

This report provides full demographic data, including military information, for the selected patient.

#### Patient Profile Medical Administration Service (MAS) (Full)

The MAS screen is the default – it specifies all dates, appointments, enrollments, team information, edits, dispositions, and the means test. The user can change the generated report to exclude particular types of information by selecting **No** for that particular type.

#### Surgery Report

This option generates a list of all available surgery reports for the selected patient. If the user selects a procedure from the list, the **OK** button is enabled.

#### Other Patient-Specific Reports

As noted below in the non-patient-specific reports section, if a veteran is selected prior to choosing Reports from the File menu, two of the reports have additional options. The **Reprint a 21-Day** **Certificate** and the **Reprint a Notice of Discharge** reports allow the user to choose either for a date or for the patient selected. Refer to the next section for details on the other reports accessed from the **Reports** option under the **File** menu.

#### Reprint a 21-Day Certificate

This option is used to reprint a 21-Day Certificate for a particular patient. After selecting a patient, the user accesses Reports under the File menu. The resulting report dialog box includes that patient as a selection option. If the user does not have a patient selected, the dialog box does not include this as an option. The certificate produced is exactly the same as the original certificate. ROC 119, which appears at the bottom of each certificate, stands for VA Form 119 – Report of Contact.

#### Reprint a Notice of Discharge

It may be occasionally necessary to reprint a Notice of Discharge for a patient. After selecting a patient, the user accesses **Reports** under the **File** menu. The resulting report dialog box includes that patient as a selection option. If the user did not have a patient selected, the dialog box would not include this as an option. If the admission associated with the 7131 was deleted and notification already sent, a message is displayed. The message includes the patient’s name, SSN, date and time of admission, notice that the admission has been deleted, and a recommendation to contact the medical center.

### CAPRI Reports (Non-Patient-Specific)

The Non-Patient-Specific CAPRI Reports are accessed by selecting **Reports** under **File** on the main menu.

#### C&P Exam Reports

##### AMIS 290

AMIS 290 reports are no longer generated from within CAPRI. When the user selects this report, they are redirected to an URL where AMIS reporting can be found. The URL displayed in CAPRI is: [http://DNS demo.DNS /dmareports.asp](http://DNS  demo.DNS   /dmareports.asp)

##### AMIS 290 by Division

AMIS 290 reports are no longer generated from within CAPRI. When the user selects this report, they are redirected to an URL where AMIS reporting can be found. The URL displayed in CAPRI is: [http://DNS demo.DNS /dmareports.asp](http://DNS  demo.DNS   /dmareports.asp)

##### Insufficient Exam Report

The **Insufficient Exam Report** option prints a report of 2507 requests entered with a priority of INSUFFICIENT EXAM for a specified date range. The user may choose a detailed or summary version of the report. Only exam reasons and types that have information to report are included on the detailed version of the report.

The summary version of the report is divided into two parts. The first part contains the total number of 2507 requests/exams received for the date range, the total number of priority insufficient requests/exams for the date range, and the percentage of insufficient requests/exams received. Due to the rounding of the component percentages, the total of the percentages may not equal 100%. The second part of the summary version is a breakdown of each reason an exam was returned. The detailed version allows the user to display one, many, or all insufficient reasons and AMIE exams.

Other information provided includes exam type, patient name, SSN, and claim number. Provider and exam date on this report is the provider and date from the originally completed 2507. The exam date is not included if the original 2507 has been purged. The length of the veteran’s name and the provider are limited to 15 characters each. If either field has been truncated, it appears with two asterisks (\*\*). If an insufficient 2507 is transferred from one site to another, that exam is reported on the insufficient exam report for both sites (original and remote).

***Please Note\*\* If Summary is chosen, CSV Comma Delimited Export will “not” be available.***

##### Pending C&P Exams

This option prints out all pending C&P requests. The user may sort the reports by request status, routing location, veteran name, or age of the request.

When sorting by status, the report can be tailored to identify NEW, PENDING, TRANSCRIBED, or ALL pending C&P exams. Sorting by TRANSCRIBED may be useful for identifying ready to rate claims that have been completed but not released to the RO. Common delays for not releasing transcribed results include waiting for a physician’s signature and/or waiting for additional required tests or studies. An inquiry to the VAMC C&P clinic can clarify reasons as to why TRANSCRIBED results have not been released if an excessive number of days have elapsed since the date of the C&P exam request.

Each report displays the following information, if applicable: veteran name, SSN, claim number, request date, elapsed days, exams requested, and requester name and location. The total number of exams pending is also provided.

##### C&P Exams Checklist

The Exam Check List for RO option is used to print a checklist used by regional office personnel to select compensation and pension examinations for veterans. The request worksheet lists the body systems and the exam worksheet names. It also contains a section for remarks. The top portion of the work sheet allows the requester to enter veteran-specific information including:

* Veteran’s name, SSN, and Claim Number (C-Number)
* VAMC where the exam is to be performed
* Veteran’s day and night telephone numbers
* Power of Attorney
* Date the exam was ordered and by whom
* Insufficient exam date

##### Re-Print C&P Final Report(s)

This option allows the reprinting of final 2507 exams with the status of **Completed, printed by RO** or **Released to RO, Not Printed**. The reports are sorted by the last two digits of the claim number. The user must enter the date the report was previously printed. Reprinting a request is not allowed unless the person requesting the reprint has a division which matches the station number of the requesting regional office. The exam must have the status **Completed, printed by RO** or **Released to RO, not printed**. The package is designed to print any lab/radiology results designated for C&P. When printing, the system examines all lab/radiology results for 120 days before the release date. The output includes a summary portion that includes patient name, SSN, claim number, and request date. The total number of requests to be printed is also provided.

##### Exam Requests by Date Range

This report generates a simple list of all exam requests entered within the specified date range. The report is sorted by date of entry. The following fields are reported: **SSN**, **Patient Name**, **Request Date**, **Date Released, Date Printed**, and **Status**. The report was created primarily to assist VHA in tracking exams that may have been requested but not released.

C&P exam Status comprises the following exam request categories:

* **New** exam requests from VBA
* **Pending, Reported** exam requests acknowledged by VHA
* Canceled by MAS
* Canceled by RO
* Released to RO, Not Printed
* Completed, Printed by RO
* New, Re-Routed
* Re-Routed, Pending Acceptance

##### Request Status by Date Range

This report generates a list of exam requests by Request Status entered within the specified date range. In addition, the statuses of **New**; **Pending, Reported**; and **Released to RO, Not Printed** may be run for a blank date range by choosing **Yes** to **Blank Date Range**. The following fields are reported: **SSN**, **Patient Name**, **Request Date**, **Date Released**, **Date Printed**, **Status**, **Date Cancelled**, **and VBA Requesting Station**.

The report is sorted by the date of the last request status change. When a blank date range is used for **New**; **Pending, Reported;** or **Released to RO** the report is sorted by request date. The report was created primarily to assist VBA to better track exam requests as they are processed.

Request Status by Date Range allows the following status filters:

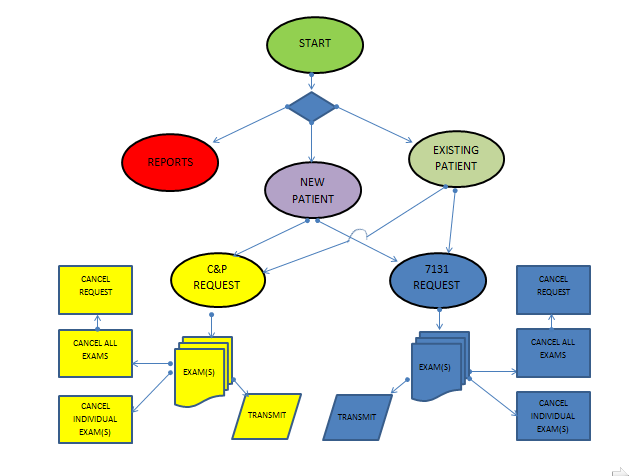
* All Statuses
* New
* Pending, Reported
* Canceled by MAS
* Canceled by RO
* Released to RO, Not Printed
* Completed, Printed by RO
* New, Re-Routed
* Re-Routed, Pending Acceptance
* Re-Routed, Accepted
* Re-Routed, Rejected

The report can also be exported to Excel in a Comma Separated Value (.CSV) file.

## Navigation Hierarchy

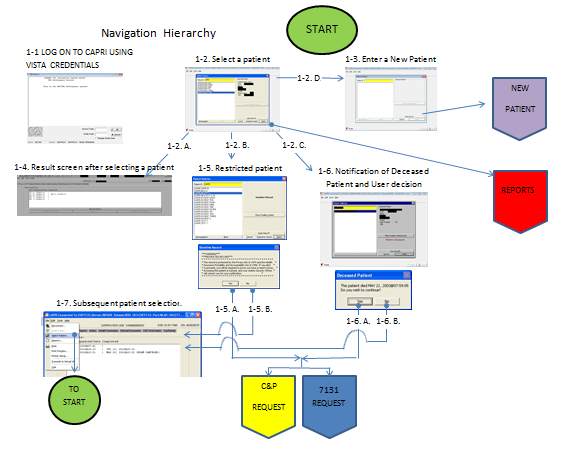
This navigation hierarchy shows how a user moves through the GUI.

Figure 8 - 25 – Navigation Heirarchy



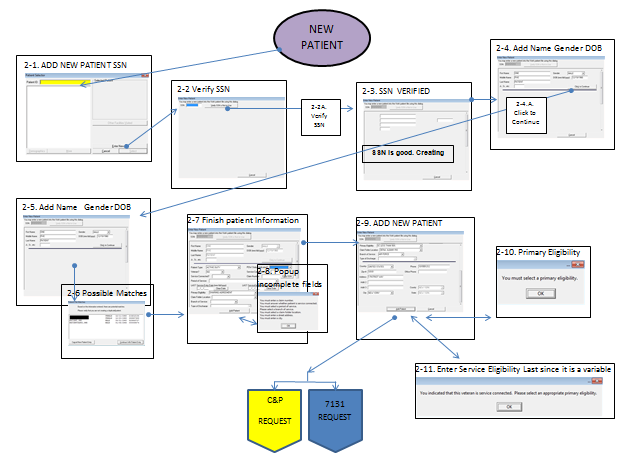
### Start – CAPRI Select a Patient

Figure 8 - 26 –Select A Patient



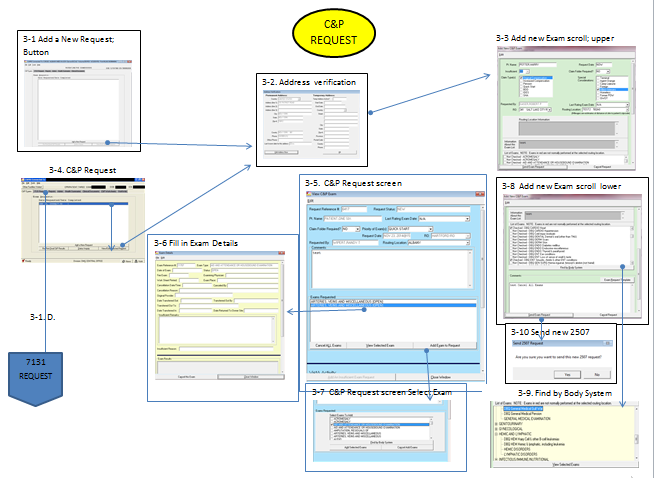
### Start – CAPRI Create a New Patient

Figure 8 - 27 – Create A New Patient



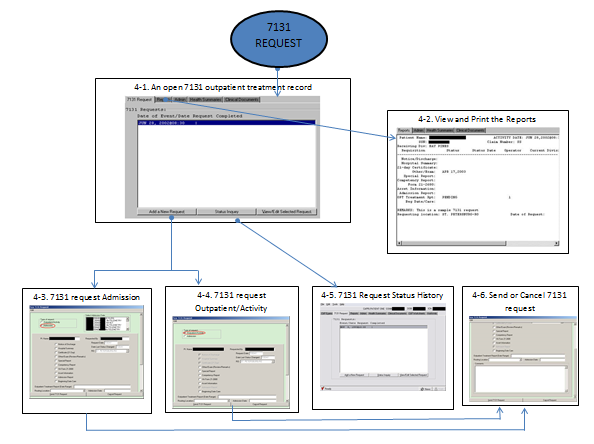
### Create a Compensation and Pension Request

Figure 8 - 28 – Create A C&P Exam Request



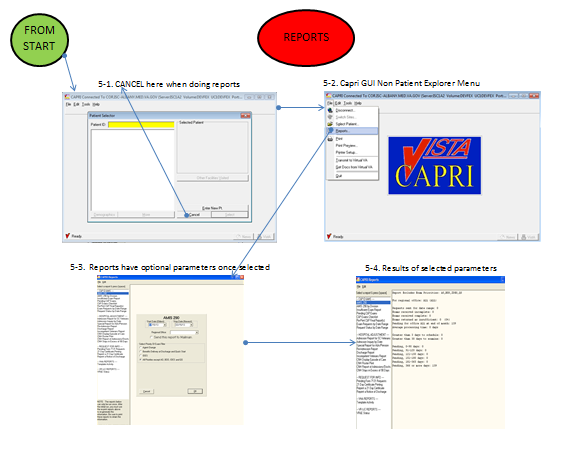
### Create a New 7131 Request

Figure 8 - 29 – Create a New 7131 Request



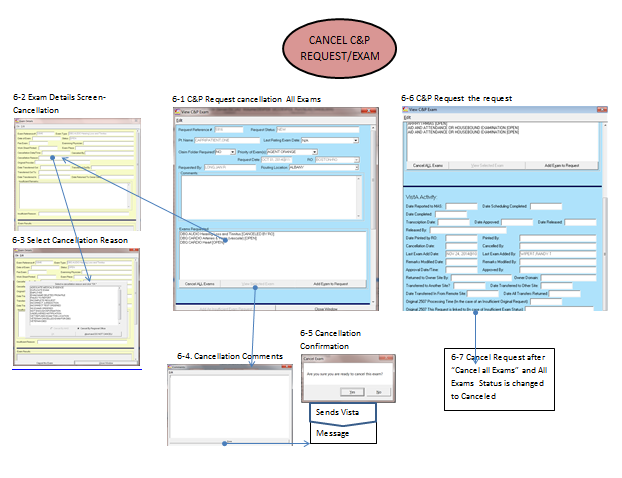
### Reports

Figure 8 - 30 - Reports



### Cancellations

Figure 8 - 31 - Cancellations



**Table 8-2 CAPRI Flow Description**

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| 1-1 | LOG ON | Log on to the Capri GUI | CAPRI | VISTA |
| 1-2 | Select a Patient | Enter a SSN or Name, If a list is presented this may be an existing pateint |  |  |
| 1-2.A | Path | Possible Result | N/A | N/A |
| 1-2.B | Path | Possible Result | N/A | N/A |
| 1-2.C | Path | Possible Path | N/A | N/A |
| 1-2.D | Path | Possible Path | N/A | N/A |
| 1-3 | Enter a New Patient | If the patient is not in the system it will follow a New Patient Entry Path | C&P Pateint Selector | CAPRI GUI Form |
| 1-4 | Result screen after selecting a patient | List of Exams for the selected patient | Search Results Tabs | CAPRI GUI |
| 1-5 | Restricted Patient | The patients records may be restricted and the user is notified and given the option to procede. | Restricted Pateint Popup | CAPRI GUI |
| 1-6 | Notification of Deceased Patient and User decision | The patient may be deceased and the user is notified and given the option to procede | Deceased Pateint Popup Message | CAPRI GUI |
| 1-5.A | Path | Possible Path to procede with restricted patient. Selecting to procede with Restricted allows for selection of Requests | N/A | N/A |
| 1-5.B | Path | Possible Path to NOT procede with restricted patient. QUITS | N/A | N/A |
| 1-6.A | Path | Possible Path to procede with Deceased patient record. Selecting to procede with Restricted allows for selection of Requests | N/A | N/A |
| 1-6.B | Path | Possible Path to NOT procede with Deceased patient record | N/A | N/A |
| 1-7 | Subsequent patient selection | Selecting to procede brings the user back here to select a patient | Pateint Selector Form | CAPRI GUI |
| 2-1 | ADD NEW PATIENT SSN | From the Figure 8 - 29, When the SSN is not already entered in the VISTA systems, the path to enter a new patient begins | Pateint Selector Form | CAPRI GUI |
| 2-2 | Verify SSN | When an SSN is entered the Verify SSN is not in Use button is available. CAPRI functions will check through Fileman Calls or Remote Procedure to see if the patient is already in the system. | SSN Verification Search Function | CAPRI GUI |
| 2-3 | SSN Verified | When the SSN is verified as being Unique; the user is notified with a popup message “SSN is good. Creating” | CAPRI Popup Message | CAPRI GUI |
| 2-4 | Add Name Gender DOB | Continue creating the patient entry by entering the basic Demographics | Enter NEW Patient | CAPRI GUI |
| 2-5 | Click to Continue | After entering the new data, “Click to continue” and CAPRI functions will check through Fileman Calls or Remote Procedure to see if the patient is already in the system again | Enter NEW Patient “Click to Continue” Button | CAPRI GUI |
| 2-6 | Possible Matches | After the search, a list of possible matches is presented to help avoid duplicate entries. | CAPRI POP up FORM | CAPRI GUI |
| 2-7 | Finish Pateint Information | After verified the status of the form changes to allow additional Demographics panels. | Enter New Patient Secondary Status WorkFlow view | CAPRI GUI |
| 2-8 | Popup Incomplete fields | The form will not save if fields are incompleted. The Popup will present a list of missing fields until all are completed | CAPRI POP up FORM | CAPRI GUI |
| 2-9 | ADD NEW PATIENT | After all necessary fields are populated and the ADD Paient button is pressed | Enter New Patient Form | CAPRI GUI |
| 2-10 | Primary Eligibility | This will popup if the primary elegibility is not selected or incorrect | CAPRI POPUP | CAPRI GUI |
| 2-11 | Enter Service Eligibility | This should be entered last. If it is selected to early and later and incompatable Primary elgibility is enterd the patient will not be ADDed. Even though the fields are populated The exact reason is not presented just this message when ADD PATIENT button is pressed. | CAPRI POPUP mesage | CAPRI GUI |
| 3-1 | Add a New Request Button | C&P Exam Add a new Reguest Tab, Select the Add a new request button | CAPRI Selection Tab form | CAPRI GUI |
| 3-2 | Address Verification | Before a New Exam is entered or an existing exam is selected the address verification screen form is presented to ensure the VA has the most current address on file | CAPRI Address Verification Form | CAPRI GUI |
| 3-3 | Add a new Exam Scroll: Upper | This form is long so scrolling is required to see the full exam request. This is the upper section | Add a New C&P Exam “Edit” form | CAPRI GUI |
| 3-4 | C&P Request | There are optional Paths from this screen depending on the Tab selected. It was placed here to demonstrate another place the Address Verification form is prompted. | C&P Request | CAPRI GUI |
| 3-5 | C&P Request Screen | Displayed after address verification, to edit or enter a request | Add a New C&P Exam “Edit” form | CAPRI GUI |
| 3-6 | Fill in Exam Details | Select an exam from the C&P Request. Double click the exam name in the Exam Request window and the Exam Details Form is presented | Add a New C&P Exam “Edit Details” form | CAPRI GUI |
| 3-7 | C&P Request screen Select Exam | When the user selects the “Add Exam to Request” button. A list of available exams list (in alphabetical order) is presented for selection. This takes the place of the previously enterd exams list that was in this field. |  | CAPRI GUI |
| 3-8 | Add a new Exam Scroll: Lower | This form is long so scrolling is required to see the full exam request. This is the lower section. | Add a New C&P Exam “Edit” form | CAPRI GUI |
| 3-9 | Find By Body System | This list is a way to narrow the search for an exam title by the Body Systems category. Select the “Find by Body System” button to bring up a list of Body Systems | Add a New C&P Exam “Edit” form option | CAPRI GUI |
| 3-10 | Send Exam Request | This verification popup is presented after the “Send Exam Request” is selected. | Add a New C&P Exam “Edit” form Pop UP | CAPRI GUI |
| 4-1 | An Open Outpatient treatment record | Highlighted is the 7131 request tab | 7131 Report Form | CAPRI GUI |
| 4-2 | View Print Reprts Tab | This is a Reports tab for the 7131 Request | Reports tab | CAPRI GUI |
| 4-3 | 7131 Request Admission | Adding a 7131 request for an “Admission” has an added field for selecting admission date | New 7131 Request Form | CAPRI GUI |
| 4-4 | 7131 Request Outpatient/Activity | Adding a 7131 request for an “Outpatient/Activity” has an no admission date field and notice of discharge, Hospital Summary;Certificate 21 Day; and Admissoin report are not available for selection | New 7131 Request Form | CAPRI GUI |
| 4-5 | 7131 Request Summary | The 7131 tab will contain a list of events and date completed. | 7131 Request Tab | CAPRI GUI |
| 4-6 | Send or Cancel Request | The lower portion of the 7131request form is where comments are entered and the request is sent. Or cancelled. | New 7131 Request Form | CAPRI GUI |
| 5-1 | Cancel here when doing reports | After logging in; the Patient Selector is automatically presented. Cancel this form to select from the File menu. | Capri Pateint Selector | CAPRI GUI |
| 5-2 | Capri GUI Non Patient Explorer Menu | Select Reports from the File Menu | CAPRI GUI File menu | CAPRI GUI |
| 5-3 | Reports have optional parameters once selected | Reports have optional parameters set by the user depending on the report selected | CAPRI GUI Reports Tab | CAPRI GUI |
| 5-4 | Results of selected parameters | One example for a selected report | CAPRI GUI Reports Tab Results Form | CAPRI GUI |
| 6-1 | C&P Request cancellation All Exams | Cancellation of a C&P request can either be individual exams of cancel all exams at once | View C&P Exams orm | CAPRI GUI |
| 6-2 | Exam Details Screen- Cancellation | Here one exam has been selected. To cancel it enter a cancellation date | Exam details form | CAPRI GUI |
| 6-3 | Select Cancellation Reason | After a Cancellation Date has been entered the user is required to select a cancellation reason | Canceation Reasons Popup Drop down list Form | CAPRI GUI |
| 6-4 | Cancellation Comments | Optional cancellation comments can be entered here | Cancellation Comments Popup | CAPRI GUI |
| 6-5 | Cancellation Confirmation | Confirming you intention to cancel the exam will cancel the exam and send a vista message containing the comments if any are made. | Cancellation Confirmation Popup | CAPRI GUI |
| 6-6 | C&P Request Cancel the request | Cancellation of the entire request occurs when the cancel all exams is selected and the cancllation date is entered in the request form | C&P Request Cancel the request | CAPRI GUI |
| 6-7 | Note: After all exams status’ are changed to Canceled you can Cancel the Request | This note is a reminder that the request cannot be cancelled until each exams’ status is changed to cancelled in the status field | View C&P exams | CAPRI GUI |

# Attachment A – Approval Signatures

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

The Business Sponsor and Project Manager are required to sign.

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

Signed: Date:

< Business Sponsor >

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

Signed: Date:

< Project Manager >

1. Additional Information
   1. Identification of Technology and Standards

The CAPRI Contract number for this project is **VA118-11-D-1010.** The Task Order number is **VA118-1010-0042**. The changes indicated in this SDD will be implemented in CAPRI Patch DVBA\*2.7\*193.

All documents and plans related to this project are located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below.

[*http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active*](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

**Table A - Definitions, Acronyms and Abbreviations**

| Term | Definition |
| --- | --- |
| ACE | Acceptable Clinical Evidence |
| A&A | Advisory & Assistance |
| AMIE | Automated Medical Information Exchange |
| AMIS | Automated Management Information System |
| BDD | Benefits Delivery at Discharge |
| BHIE | Bidirectional Health Information Exchange ([VA](http://vista.med.DNS   /iss/acronyms/v-acronyms.asp#va) and [DoD](http://vista.med.DNS   /iss/acronyms/d-acronyms.asp#dod)) |
| BP | Blood Pressure |
| C&A | Certification and Accreditation |
| C&P | Compensation and Pension |
| CAPRI | Compensation and Pension Record Interchange |
| CPEP | Compensation and Pension Examination Program |
| CPRS | Computerized Patient Record System |
| CCR | CAPRI Contractor Referral |
| DBQ | Disability Benefits Questionnaire |
| DCS | DES Claimed Condition By Service Member |
| demTRAN | Disability Examination Management Tracking, Referral and Notification Utility |
| DES | Disability Evaluation System |
| DFD | DES Fit for Duty |
| DLL | Dynamic Link Library |
| DoD | Department of Defense |
| DVBA | The pre-fix for AUTOMATED MED INFO EXCHANGE (namespace). VBA's interface into VistA. |
| ESM | Enterprise System Management |
| FHIE | Federal Health Information Exchange |
| FISMA | Federal Information Security Management Act |
| GB | Gigabyte |
| GUI | Graphical User Interface |
| HIA | Health Information Architecture |
| HRC | Health Revenue Center |
| HTML | HyperText Markup Language |
| IDES | Integrated Disability Evaluation System |
| MAS | Medical Administration Service |
| MVI | Master Veteran Index |
| OI | Office of Information |
| OED | Office of Enterprise Development |
| OHI | Office of Health Information |
| OMB | Office of Management and Budget |
| PWS | Performance Work Statement |
| PNCS | Progress Note Construction Set |
| RAM | Random Access Memory |
| RAEM | Requirements Analysis and Engineering Management |
| RPC | Remote Procedure Call |
| RSD | Requirements Specification Document |
| SDD | Software Design Document |
| SQA | Software Quality Assurance |
| SSH | Secure Shell |
| SSO | Single Sign-On |
| TIU | Text Integration Utilities |
| URL | Universal Resource Locator |
| VA | Department of Veterans Affairs |
| VBA | Veterans Benefits Administration |
| VHA | Veterans Health Administration |
| VistA | Veterans Health Information Systems and Technology Architecture |
| XREF | Cross Reference |

* 1. Constraining Policies, Directives and Procedures

The System Design Document follows at a minimum the constraints and requirements of the ProPath template for this document. This ProPath template has been expanded to include additional content, specifically more detailed system and hardware architecture sections. This document conforms to the 508 policy for VA.

* 1. Requirements Traceability Matrix

All documents and plans related to this project are located in the Active Project Notebook found in the Technical Services Project Repository (TSPR). The direct link is listed below.

[*http://DNS .med.DNS /warboard/anotebk.asp?proj=1702&Type=Active*](http://DNS       .med.DNS   /warboard/anotebk.asp?proj=1702&Type=Active)

* 1. Packaging and Installation

N/A

* 1. Design Metrics

None