Computerized Patient Record System (CPRS) v32

System Design Document



August 2017

Version 1.35

Department of Veterans Affairs

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/3/2007 | 1.37 | Added new routine PSSHRQ26 for NSR#20080704 | Andrea Freeman |
| 11/2/2017 | 1.36 | Added comments for Park a Prescription changes. These changes will not allow the medication Clozapine to be ‘Parked’ and not generate ECME claims on ‘Parked’ medications. Add routine PSB82PST for NSR#20110903. | Andrea Freeman |
| 08/03/2017 | 1.35 | Updated XQALDATA routine entry and added XQALDEL and XU8P662 routine entries for NSR 20081008. Also added details for the new PAR cross-ref for the Alert Tracking file (#8992.1) in the database repository section for the same NSR. | Jeff Swesky |
| 06/13/2017 | 1.34 | Added routine table for PSSPOIM1 and updated Modified Logic section under PSSPOIMO for PSS\*1.0\*204. Updated Modified Logic sections for GMTSLRS, GMTSLRSC, and GMTSLRME for corrected issue tracker defects. | Jeff Swesky |
| 05/30/2017 | 1.33 | Added changes to PSSHRQ21 | Andrea Freeman |
| 04/27/2017 | 1.32 | Added additional tag to PSSHREQ | Andrea Freeman |
| 03/31/2017 | 1.31 | Updated TIUSRVLO and added TIUSRVLI and ORCNOTE for NSR #20070817  Added VistA changes for PSPO #934 | Ty Phelps |
| 03/10/2017 | 1.30 | Updated ORWDAL32 and added GMRAPEM0 for NSR #20070203  Added additional routine for Park a Prescripton, PSODIR3 | Jeff Swesky  Andrea Freeman |
| 02/23/2017 | 1.29 | Added routines ORWDPS2 and PSS50P7. Also, added changes for Pharmacy Orderable Items (#50.7).  Added ^TMP global for COTS processing | Donna Sandusky  Andrea Freeman |
| 01/27/2017 | 1.28 | Added NSR #20100101 Indication for Use  Added updates to COTS routines | Donna Sandusky  Andrea Freeman |
| 01/24/2017 | 1.27 | Included some rearrangement. | Scott Stephenson  Jamie Crumley |
| 12/1/2016 | 1.26 | Added OR, LR, and GMTS routines modified for NSR 20081206. Added routines XUAF4 and XUMF04H and new fields for multiple #999.1 of file #4 for NSR 20081206.  Added new routine ORDV06D and unique records for file 101.24 for NSR 20070203. Removed ORDV06 entry for NSR 20070203, since ORDV06D is used in its place.  Added routine ORCACT0 and a new Security Key multiple for the Display Group file for NSR 20071103. | Jeff Swesky |
| 11/16/2016 | 1.25 | Worked on 508 compliance. General editing . | Jamie Crumley |
| 9/22/16 | 1.24 | Added additional changes to PSSHREQ for NSR#20080704 | Andrea Freeman |
| 8/31/2016 | 1.23 | Monthly Updates | Rishan Chandarana |
| 7/30/2016 | 1.22 | Added dictionary changes to Data Entries Affected by the Design section  Added info for NSR 20110210, Separate Alert for Prosthetics Requests  Added Description of implementation of NSR 20070817 Progress Notes Display Misleading | Andrea Freeman  William A. Thompson  Andrey Andriyevskiy |
| 7/5/2016 | 1.21 | Updated routines PSSHREQ, PSSHRIT and PSSHRQ23 (held for future modifications) | Andrea Freeman |
| 5/25/2016 | 1.20 | Updated Alt text for images in section 6.2.2.3.15 (GUI: Detailed Design|Softrware Detailed Design|Specific Requirements|Design Element Tables|GUI).  Added ALLERGY tag to PSSHREQ. | Andrey Andriyevskiy  Andrea Freeman |
| 4/11/2016 | 1.19 | Updated routines PSBALL, PSBOAL, and PSBODO (added tags ORCHECK, ALLERGY) | Andrea Freeman |
| 3/25/2016 | 1.18 | Added PSBOAL (tags EN1, OBS, COM) and PSBALL (tag SORT) and PSBODO (adding new tag DSPORCK and modifying tag DISPORD) for NSR#201000825 | Andrea Freeman |
| 2/29/2016 | 1.17 | Added PSODRG (tag SET) PSSHRIT (tags TESTS & QTESTS) | Andrea Freeman |
| 1/28/2016 | 1.16 | Added PSSHREQ – with new tag CHKALLER  Added Routine and RPC elements for 20081206, 20081008 and 20070203  Added routine PSSHRQ23 intial design. | Andrea Freeman, Robert Lauro |
| 01/04/15 | 1.15 | Updated Routine and RPC section with changes for NSRs 20081008, 20070203, 20070811, 20101203 and 20071211 | Robert Lauro |
| 12/3/15 | 1.14 | Updated the Routine section for ORQ2 with additional coding | Andrea Freeman |
| 10/29/15 | 1.13 | Updated to new template; updated for NSRs 20110719, 20071211, and 20101203 | Jamie Crumley, Kim Hovorka |
| 9/30/15 | 1.12 | Updated Routine section under Park A Prescription. Added routines: ORWORR, PSO52EX, ORWPS1 | Andrea Freeman |
| 8/31/2015 | 1.11 | Updated the Routine section for routine ORQ2 | Andrea Freeman |
| 8/3/2015 | 1.10 | Add text for AERB to section 1.5 & 6.1, update 20071211 | Andrea Freeman, Rishan Chandarana |
| 7/6/2015 | 1.9 | Monthly Updates | Craig Hinton |
| 5/19/2015 | 1.8 | Update NSR 20110903, and 20080226. | Jamie Crumley, Kim Hovorka |
| 5/1/2015 | 1.7 | Update in general in response to AERB | Jamie Crumley, Rishan Chandarana |
| 4/6/2015 | 1.6 | Update sections 2,3 | Rishan Chandarana |
| 3/2/15 | 1.5 | Add section for Park A Prescription routines; Updated sections 1,2,6 | Andrea Freeman, Ty Phelps, Mike Jenkins, Kim Hovorka, Craig Hinton, Rishan Chandarana |
| 2/5/2015 | 1.4 | Made progress to conform to AERB direction; updated content related to NSR 20110606 | Mike Jenkins, Kim Hovorka |
| 12/31/2014 | 1.3 | Added code changes for Discontinue Order on Adverse Reaction. | Steve Graham |
| 11/30/2014 | 1.2 | Updated with screen shots for Similar Provider Names. Added implementation detail for Discontinue Order on Adverse Reaction. | Kim Hovorka , Ty Phelps, Steve Graham |
| 10/31/2014 | 1.1 | Editorial changes by Technical Writer. | Ty Phelps, Blair Sanders, Craig Hinton |
| 9/19/2014 | 1.0 | Initial Design | Ty Phelps, Jamie Crumley, Rishan Chandarana |

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.

Instructions

| Activity | New Capability (1) | Feature Enhancement (2) |
| --- | --- | --- |
| **Field Deployment (A)** | Yes | Yes |
| **Cloud/Web Deployment (B)** | Yes | Yes |
| **Mobile Application (C)** | Yes | Yes |

Table of Contents

[1. Introduction 1](#_Toc475621120)

[1.1. Scope 1](#_Toc475621121)

[1.2. User Profiles 1](#_Toc475621122)

[2. Background 1](#_Toc475621123)

[2.1. Overview of the System 1](#_Toc475621124)

[2.2. Overview of the Business Process 2](#_Toc475621125)

[2.3. Overview of the Significant Requirements 3](#_Toc475621126)

[2.3.1. Drug-Allergy Order Check Enhancements and Improved Detail Request (#20100825) (RSD #2.6.1) 3](#_Toc475621127)

[2.3.2. Changes to the Nurse Order Verification’s Effect on Order Status Request (#20090416) (RSD #2.6.2) 3](#_Toc475621128)

[2.3.3. Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216) (RSD #2.6.3) 3](#_Toc475621129)

[2.3.4. CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4) 4](#_Toc475621130)

[2.3.5. Add Address of Performing Lab to Reports in CPRS Health Summary Request (#20081206) (RSD #2.6.6) 4](#_Toc475621131)

[2.3.6. CPRS Day of the Week Med Schedule Change Request (#20120802) (RSD #2.6.7) 4](#_Toc475621132)

[2.3.7. Prevent Confusion over CPRS Status Display of Orders and Available Actions Request (#20111006) (RSD #2.6.8) 4](#_Toc475621133)

[2.3.8. Change in Unflagging Capabilities Request (#20071103) (RSD #2.6.9) 4](#_Toc475621134)

[2.3.9. CPRS Notification Alert Processing Improvement Request (#20081008) (RSD #2.6.10) 5](#_Toc475621135)

[2.3.10. Confirm Provider Selected with Similar Names Request (#20110606) (RSD #2.6.11) 5](#_Toc475621136)

[2.3.11. Identify Required Fields in TIU Note Templates Request (#20100706) (RSD #2.6.12) 5](#_Toc475621137)

[2.3.12. Adverse Reaction Reporting File Modification Request (#20120404) (RSD #2.6.13) 5](#_Toc475621138)

[2.3.13. Allergy Order Check Enhancement Request (#20070203) (RSD #2.6.14) 5](#_Toc475621139)

[2.3.14. Park-A-Prescription (#20090509) (RSD #2.6.15) 6](#_Toc475621140)

[2.3.15. First Dose – Enhancement within CPRS Medication Order Request (#20070811) (RSD #2.6.16) 6](#_Toc475621141)

[2.3.16. IV Administration Not Documented in CPRS Request (#20110903) (RSD #2.6.17) 6](#_Toc475621142)

[2.3.17. Critical / High Order Check Display Request (#20101203) (RSD #2.6.18) 6](#_Toc475621143)

[2.3.18. Button to Link No Assessment Warning to Allergy Assessment Screens Request (#20070920) (RSD #2.6.19) 7](#_Toc475621144)

[2.3.19. Real-time Notification of Potentially Missed Order Checks Request(#20060710) (RSD #2.6.20) 7](#_Toc475621145)

[2.3.20. Enhanced Allergy Checks Using COTS Data Request (#20080704) (RSD #2.6.21) 7](#_Toc475621146)

[2.3.21. D/C Order by Adverse Reaction Request (#20080226) (RSD #2.6.22) 8](#_Toc475621147)

[2.3.22. Changes to Allergy/Pharmacy Packages Request (#20071211) (RSD #2.6.23) 8](#_Toc475621148)

[2.3.23. Order Flag Recommendations Request (#20110719) (RSD #2.6.24) 8](#_Toc475621149)

[2.3.24. Progress Notes Display Misleading Request (#20070817) (RSD #2.6.25) 8](#_Toc475621150)

[2.3.25. Indication for Prescription and Medication Orders (#20100101) (RSD #2.6.26) 8](#_Toc475621151)

[3. Conceptual Design 8](#_Toc475621152)

[3.1. Conceptual Application Design 8](#_Toc475621153)

[3.1.1. Application Context 17](#_Toc475621154)

[3.1.2. High-Level Application Design 18](#_Toc475621155)

[3.1.3. Application Locations 19](#_Toc475621156)

[3.2. Conceptual Data Design 20](#_Toc475621157)

[3.2.1. Project Conceptual Data Model 20](#_Toc475621158)

[3.2.2. Database Information 20](#_Toc475621159)

[3.2.3. User Interface Data Mapping 20](#_Toc475621160)

[3.2.3.1. Application Screen Interface 20](#_Toc475621161)

[3.2.3.1.1. Discontinue / Cancel Orders 20](#_Toc475621162)

[3.2.3.1.2. List of allergies currently recorded for the patient 22](#_Toc475621163)

[3.2.3.1.3. Similar Provider Screen 22](#_Toc475621164)

[3.2.3.1.4. Unflagging Orders Screen 23](#_Toc475621165)

[3.2.3.1.5. Allergy Pharmacy Packages Check Screen 24](#_Toc475621166)

[3.2.3.1.6. Order Flagging Screens 24](#_Toc475621167)

[3.2.3.2. Application Report Interface 26](#_Toc475621168)

[3.2.3.2.1. Add Performing Lab to Report 26](#_Toc475621169)

[3.2.3.3. Unmapped Data Element 26](#_Toc475621170)

[3.3. Conceptual Infrastructure Design 26](#_Toc475621171)

[3.3.1. System Criticality and High Availability 27](#_Toc475621172)

[3.3.2. Special Technology 27](#_Toc475621173)

[3.3.3. Technology Locations 27](#_Toc475621174)

[3.3.4. Conceptual Infrastructure Diagram 27](#_Toc475621175)

[3.3.4.1. Location of Environments and External Interfaces 27](#_Toc475621176)

[3.3.4.2. Conceptual Production String Diagram 28](#_Toc475621177)

[4. System Architecture 28](#_Toc475621178)

[4.1. Hardware Architecture 28](#_Toc475621179)

[4.2. Software Architecture 29](#_Toc475621180)

[4.3. Network Architecture 29](#_Toc475621181)

[4.4. Service Oriented Architecture / ESS 30](#_Toc475621182)

[4.5. Enterprise Architecture 30](#_Toc475621183)

[5. Data Design 30](#_Toc475621184)

[5.1. DBMS Files 30](#_Toc475621185)

[5.2. Non-DBMS Files 30](#_Toc475621186)

[5.3. Data View 31](#_Toc475621187)

[6. Detailed Design 31](#_Toc475621188)

[6.1. Hardware Detailed Design 31](#_Toc475621189)

[6.2. Software Detailed Design 31](#_Toc475621190)

[6.2.1. Conceptual Design 31](#_Toc475621191)

[6.2.1.1. Product Perspective 31](#_Toc475621192)

[6.2.1.1.1. User Interfaces 31](#_Toc475621193)

[6.2.1.1.2. Hardware Interfaces 32](#_Toc475621194)

[6.2.1.1.3. Software Interfaces 32](#_Toc475621195)

[6.2.1.1.4. Communications Interfaces 32](#_Toc475621196)

[6.2.1.1.5. Memory Constraints 32](#_Toc475621197)

[6.2.1.1.6. Special Operations 32](#_Toc475621198)

[6.2.1.2. Product Features 32](#_Toc475621199)

[6.2.1.3. User Characteristics 32](#_Toc475621200)

[6.2.1.4. Dependencies and Constraints 32](#_Toc475621201)

[6.2.2. Specific Requirements 33](#_Toc475621202)

[6.2.2.1. Database Repository 33](#_Toc475621203)

[6.2.2.2. System Features 34](#_Toc475621204)

[6.2.2.2.1. NSR # 20070203 - Allergy Order Check Enhancement 34](#_Toc475621205)

[6.2.2.2.1.1. ORWDAL32 34](#_Toc475621206)

[6.2.2.2.1.2. ORDV06D 37](#_Toc475621207)

[6.2.2.2.1.3. RPC ORWDAL32 CHKMEDS 40](#_Toc475621208)

[6.2.2.2.1.4. RPC ORWDAL32 GETPROV 40](#_Toc475621209)

[6.2.2.2.1.5. RPC ORWDAL32 GETPROV 41](#_Toc475621210)

[6.2.2.2.2. NSR #20080704 - Enhanced Allergy Checks COTS 41](#_Toc475621211)

[6.2.2.2.2.1. PSODRG 41](#_Toc475621212)

[6.2.2.2.2.2. PSSHREQ 42](#_Toc475621213)

[6.2.2.2.2.3. PSSHREQ 45](#_Toc475621214)

[6.2.2.2.2.4. PSSHRIT 47](#_Toc475621215)

[6.2.2.2.2.5. PSSHRQ20 50](#_Toc475621216)

[6.2.2.2.2.6. COTS ALLERGY MAPPING (#120.89 -- ^GMRD(120.89,)) 52](#_Toc475621217)

[6.2.2.2.3. NSR # 20081008 - CPRS Notification Alert Processing 52](#_Toc475621218)

[6.2.2.2.3.1.1. ORWORB 54](#_Toc475621219)

[6.2.2.2.3.1.2. ORWTPR 55](#_Toc475621220)

[6.2.2.2.4. NSR 20081206 – Add Address of Performing Lab to Reports in CPRS Health Summary 57](#_Toc475621221)

[6.2.2.2.4.1. GMTSLRC 57](#_Toc475621222)

[6.2.2.2.4.2. GMTSLRCE 59](#_Toc475621223)

[6.2.2.2.4.3. GMTSLRSC 60](#_Toc475621224)

[6.2.2.2.4.4. GMTSLRS7 61](#_Toc475621225)

[6.2.2.2.4.5. GMTSLRCP 63](#_Toc475621226)

[6.2.2.2.4.6. GMTSLRPE 64](#_Toc475621227)

[6.2.2.2.4.7. GMTSLRM 65](#_Toc475621228)

[6.2.2.2.4.8. GMTSLRME 66](#_Toc475621229)

[6.2.2.2.4.9. GMTSLRA 67](#_Toc475621230)

[6.2.2.2.4.10. GMTSLRAE 68](#_Toc475621231)

[6.2.2.2.4.11. GMTSLRS 69](#_Toc475621232)

[6.2.2.2.4.12. GMTSLRSE 70](#_Toc475621233)

[6.2.2.2.4.13. XUMF04H 71](#_Toc475621234)

[6.2.2.2.4.14. XUAF4 73](#_Toc475621235)

[6.2.2.2.4.15. ORWLR1 75](#_Toc475621236)

[6.2.2.2.4.16. ORWRPL 75](#_Toc475621237)

[6.2.2.2.4.17. ORWRPP 78](#_Toc475621238)

[6.2.2.2.4.18. ORWRPP1 79](#_Toc475621239)

[6.2.2.2.4.19. LR7OB63 80](#_Toc475621240)

[6.2.2.2.4.20. LR7OGG 82](#_Toc475621241)

[6.2.2.2.4.21. LR7OGMG 83](#_Toc475621242)

[6.2.2.2.4.22. LR7OR2 83](#_Toc475621243)

[6.2.2.2.4.23. LR7OSAP 85](#_Toc475621244)

[6.2.2.2.4.24. LR7OSUM2 85](#_Toc475621245)

[6.2.2.2.4.25. LRRPL 87](#_Toc475621246)

[6.2.2.2.4.26. GMTSLRMB 88](#_Toc475621247)

[6.2.2.2.4.27. GMTSLROE 89](#_Toc475621248)

[6.2.2.2.4.28. GMTSLROB 90](#_Toc475621249)

[6.2.2.2.4.29. GMTSLROS 91](#_Toc475621250)

[6.2.2.2.4.30. GMTSLREE 93](#_Toc475621251)

[6.2.2.2.4.31. GMTSLREM 94](#_Toc475621252)

[6.2.2.2.4.32. ORDV02 95](#_Toc475621253)

[6.2.2.2.5. NSR# 20100706 (Identify Required Fields in Text Integration Utility (TIU) Note Templates) 97](#_Toc475621254)

[6.2.2.2.5.1. ORWTIU 97](#_Toc475621255)

[6.2.2.2.6. NSR# 20071211 – Changes to Allergy Pharmacy Package 101](#_Toc475621256)

[6.2.2.2.6.1. ORWDXC 101](#_Toc475621257)

[6.2.2.2.6.2. ORKCHK 103](#_Toc475621258)

[6.2.2.2.6.3. ORKCHK5 103](#_Toc475621259)

[6.2.2.2.6.4. ORCSAVE 105](#_Toc475621260)

[6.2.2.2.6.5. ORCSAVE2 106](#_Toc475621261)

[6.2.2.2.7. NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail 113](#_Toc475621262)

[6.2.2.2.7.1. PSBOAL 113](#_Toc475621263)

[6.2.2.2.7.2. PSBALL 116](#_Toc475621264)

[6.2.2.2.7.3. PSODO 118](#_Toc475621265)

[6.2.2.2.7.4. PSODO 121](#_Toc475621266)

[6.2.2.2.8. NSR 20071103 – Change in Unflagging Capabilities 124](#_Toc475621267)

[6.2.2.2.8.1. ORCACT0 124](#_Toc475621268)

[6.2.2.2.9. NSR 20110210 – Create Separate Alert for Prosthetics Requests Screening Call Record 125](#_Toc475621269)

[6.2.2.2.9.1. GMRCP 125](#_Toc475621270)

[6.2.2.2.9.2. GMRAGUI1 127](#_Toc475621271)

[6.2.2.2.9.3. ORWDAL32 128](#_Toc475621272)

[6.2.2.2.9.4. ORWDX2 128](#_Toc475621273)

[6.2.2.2.9.5. PSJHL5 129](#_Toc475621274)

[6.2.2.2.9.6. ORDV06 130](#_Toc475621275)

[6.2.2.2.9.7. ORQ21 131](#_Toc475621276)

[6.2.2.2.9.8. ORWDPS1 132](#_Toc475621277)

[6.2.2.2.9.9. TIUSRVL0 133](#_Toc475621278)

[6.2.2.2.9.10. ORBPRCHK 134](#_Toc475621279)

[6.2.2.2.10. NSR# 20090509 (15) – Park a Prescription 135](#_Toc475621280)

[6.2.2.2.11. NSR# 20080307 CWAD Post Auto-Demotion Rules Request (RSD #2.6.4) 142](#_Toc475621281)

[6.2.2.2.11.1. Routine TIUCWD 142](#_Toc475621282)

[6.2.2.2.11.2. Routine TIUCWDUT 144](#_Toc475621283)

[6.2.2.2.12. NSR# 20110903 - IV infusion administration not accurately documented in CPRS logs 147](#_Toc475621284)

[6.2.2.2.12.1. Routine PSBOPM 148](#_Toc475621285)

[6.2.2.2.13. NSR# 20071211 Changes to Allergy/Pharmacy Packages 149](#_Toc475621286)

[6.2.2.2.14. NSR # 200120404 – Adverse Reaction Reporting File Modifications 157](#_Toc475621287)

[6.2.2.2.14.1. Routine GMRADPT 157](#_Toc475621288)

[6.2.2.2.14.2. Routine GMRAGUI 159](#_Toc475621289)

[6.2.2.2.14.3. Routine GMRAGUI1 160](#_Toc475621290)

[6.2.2.2.14.4. Routine GMRAOR 162](#_Toc475621291)

[6.2.2.2.14.5. Routine GMRAOR1 166](#_Toc475621292)

[6.2.2.2.14.6. Routine GMRAOR2 168](#_Toc475621293)

[6.2.2.2.14.7. Routine GMRAPEH0 171](#_Toc475621294)

[6.2.2.2.14.8. Routine GMRAPEM1 172](#_Toc475621295)

[6.2.2.2.14.9. Routine GMRAPER0 173](#_Toc475621296)

[6.2.2.2.14.10. Routine ORQQAL 174](#_Toc475621297)

[6.2.2.2.15. Templates 176](#_Toc475621298)

[6.2.2.2.16. Bulletins 176](#_Toc475621299)

[6.2.2.2.17. Data Entries Affected by the Design 177](#_Toc475621300)

[6.2.2.2.17.1. CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4) 177](#_Toc475621301)

[6.2.2.2.17.1.1. TIU PERSONAL PREFERENCES (#8926) 177](#_Toc475621302)

[6.2.2.2.17.2. NSR #20110210 Create Separate Alert for Prosthetics Requests Screening Call Record 178](#_Toc475621303)

[6.2.2.2.17.2.1. OE/RR NOTIFICATIONS (#100.9) 178](#_Toc475621304)

[6.2.2.2.17.3. NSR #20090509 (15) – Park a Prescription 178](#_Toc475621305)

[6.2.2.2.17.3.1. PRESCRIPTION FILE (#52) 178](#_Toc475621306)

[6.2.2.2.17.3.2. OUTPATIENT SITE FILE (#59) 179](#_Toc475621307)

[6.2.2.2.17.3.3. PENDING OUTPATIENT ORDERS FILE (#52.41) 179](#_Toc475621308)

[6.2.2.2.17.4. NSR # 20071211 (23) – Changes to Allergy/Pharmacy Packages 179](#_Toc475621309)

[6.2.2.2.17.4.1. ORDER CHECK INSTANCES FILE (#100.05) 179](#_Toc475621310)

[6.2.2.2.17.4.2. ORDER CHECK OVERRIDE REASONS (#100.04 -- ^ORD(100.04)) 180](#_Toc475621311)

[NSR # 20100101 (26) – Indications for Use for Prescription and Medication Orders 180](#_Toc475621312)

[Pharmacy Orderable Items (#50.7) 180](#_Toc475621313)

[6.2.2.2.18. Unique Record(s) 181](#_Toc475621314)

[6.2.2.2.18.1. D/C Order by Adverse Reaction Request #20080226 181](#_Toc475621315)

[6.2.2.2.18.2. Add Address of Performing Lab to Reports in CPRS Health Summary (#20081206) 181](#_Toc475621316)

[6.2.2.2.18.3. CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4) 181](#_Toc475621317)

[6.2.2.2.19. File or Global Size Changes 181](#_Toc475621318)

[6.2.2.2.20. Mail Groups 182](#_Toc475621319)

[6.2.2.2.21. Security Keys 182](#_Toc475621320)

[6.2.2.2.22. Options 183](#_Toc475621321)

[6.2.2.2.23. Protocols 184](#_Toc475621322)

[6.2.2.2.24. Remote Procedure Call (RPC) 185](#_Toc475621323)

[6.2.2.2.25. Constants Defined in Interface 186](#_Toc475621324)

[6.2.2.2.26. Variables Defined in Interface 187](#_Toc475621325)

[6.2.2.2.27. Types Defined in Interface 187](#_Toc475621326)

[6.2.2.2.28. GUI 187](#_Toc475621327)

[6.2.2.2.28.1. Modify the Similar Patients form to also display similar providers and cosigners (NSR#20110606) 189](#_Toc475621328)

[6.2.2.2.28.1.1. Summary of changes 189](#_Toc475621329)

[6.2.2.2.28.1.2. Files changed 189](#_Toc475621330)

[6.2.2.2.28.1.3. Unit fAddlSigners 190](#_Toc475621331)

[6.2.2.2.28.1.4. Unit fAlertForward 192](#_Toc475621332)

[6.2.2.2.28.1.5. Unit fDCSummProps 193](#_Toc475621333)

[6.2.2.2.28.1.6. Unit fDupPts 197](#_Toc475621334)

[6.2.2.2.28.1.7. Unit fEncnt 200](#_Toc475621335)

[6.2.2.2.28.1.8. Unit fNoteCPFields 204](#_Toc475621336)

[6.2.2.2.28.1.9. Unit fNoteProps 205](#_Toc475621337)

[6.2.2.2.28.1.10. Unit fPtSel 207](#_Toc475621338)

[6.2.2.2.28.1.11. Unit fODConsult 208](#_Toc475621339)

[6.2.2.2.28.1.12. Unit fODProc 209](#_Toc475621340)

[6.2.2.2.28.1.13. Unit fOrdersAlert 210](#_Toc475621341)

[6.2.2.2.28.1.14. Unit fPtSelOptns 212](#_Toc475621342)

[6.2.2.2.28.1.15. Unit rCore 213](#_Toc475621343)

[6.2.2.2.28.1.16. Unit uEditConsult 214](#_Toc475621344)

[6.2.2.2.28.1.17. Unit fVisitType 215](#_Toc475621345)

[6.2.2.2.28.1.18. Unit uSimilarNames 216](#_Toc475621346)

[6.2.2.2.28.1.19. Unit CPRSChart.dpr 224](#_Toc475621347)

[6.2.2.2.28.2. Update Surrogate Management Functionality within CPRS GUI (20071216) 225](#_Toc475621348)

[6.2.2.2.28.2.1. Summary of changes 225](#_Toc475621349)

[6.2.2.2.28.2.2. Files modified 225](#_Toc475621350)

[6.2.2.2.28.2.3. GUI Classes 226](#_Toc475621351)

[6.2.2.2.28.2.4. File CPRSChart.dpr changes 226](#_Toc475621352)

[6.2.2.2.28.2.5. File fOptions.pas changes 226](#_Toc475621353)

[6.2.2.2.28.2.6. File fOptionsSurrogate.pas 231](#_Toc475621354)

[6.2.2.2.28.2.6.1. Components on Form TfrmOptionsSurrogate 231](#_Toc475621355)

[6.2.2.2.28.2.6.2. Event Handlers of form TfrmOptionsSurrogate 232](#_Toc475621356)

[6.2.2.2.28.2.6.3. Class Methods (form TfrmOptionsSurrogate) 237](#_Toc475621357)

[6.2.2.2.28.2.6.4. Special References 247](#_Toc475621358)

[6.2.2.2.28.2.6.5. Class Fields (form TfrmOptionsSurrogate) 247](#_Toc475621359)

[6.2.2.2.28.2.6.6. Class Properties (form TfrmOptionsSurrogate) 247](#_Toc475621360)

[6.2.2.2.28.2.6.7. Unit Variables 247](#_Toc475621361)

[6.2.2.2.28.2.6.8. Unit Constants 248](#_Toc475621362)

[6.2.2.2.28.2.6.9. Unit Methods 248](#_Toc475621363)

[6.2.2.2.28.2.6.10. Additional classes defined in the unit 250](#_Toc475621364)

[6.2.2.2.28.2.6.11. Class TIdItem Fields 250](#_Toc475621365)

[6.2.2.2.28.2.6.12. Class TIdItem Methods 251](#_Toc475621366)

[6.2.2.2.28.2.7. File fSurrogateEdit.pas changes 253](#_Toc475621367)

[6.2.2.2.28.2.7.1. Components on Form 253](#_Toc475621368)

[6.2.2.2.28.2.7.2. Events 254](#_Toc475621369)

[6.2.2.2.28.2.7.3. Class Methods 254](#_Toc475621370)

[6.2.2.2.28.2.7.4. Special References 254](#_Toc475621371)

[6.2.2.2.28.2.7.5. Class Events 255](#_Toc475621372)

[6.2.2.2.28.2.7.6. Class Fields 255](#_Toc475621373)

[6.2.2.2.28.2.7.7. Class Properties 255](#_Toc475621374)

[6.2.2.2.28.2.7.8. Unit Variables 255](#_Toc475621375)

[6.2.2.2.28.2.7.9. Unit Methods 255](#_Toc475621376)

[6.2.2.2.28.2.8. File uConst.pas changes 256](#_Toc475621377)

[6.2.2.2.28.2.8.1. Components on Form 256](#_Toc475621378)

[6.2.2.2.28.2.8.2. Events 256](#_Toc475621379)

[6.2.2.2.28.2.8.3. Special References 256](#_Toc475621380)

[6.2.2.2.28.2.8.4. Class Events 256](#_Toc475621381)

[6.2.2.2.28.2.8.5. Unit Methods 256](#_Toc475621382)

[6.2.2.2.28.2.8.6. Class Properties 256](#_Toc475621383)

[6.2.2.2.28.2.8.7. Uses Clause 256](#_Toc475621384)

[6.2.2.2.28.2.9. File rOptions.pas changes 257](#_Toc475621385)

[6.2.2.2.28.2.10. File ORDtTmCal.pas changes 258](#_Toc475621386)

[6.2.2.2.28.2.10.1. Uses Clause changes 258](#_Toc475621387)

[6.2.2.2.28.2.10.2. Components Modified 258](#_Toc475621388)

[6.2.2.2.28.2.11. File ORDtTm.pas changes 261](#_Toc475621389)

[6.2.2.2.28.2.11.1. Uses Clause changes 261](#_Toc475621390)

[6.2.2.2.28.2.11.2. Components modified 261](#_Toc475621391)

[6.2.2.2.28.3. Identify Required Fields in Text Integration Utility (TIU) Note Templates (20100706) 276](#_Toc475621392)

[6.2.2.2.28.3.1. Summaryof changes 276](#_Toc475621393)

[6.2.2.2.28.3.2. Files modified 276](#_Toc475621394)

[6.2.2.2.28.3.3. File fOptions.pas changes 278](#_Toc475621395)

[6.2.2.2.28.3.4. File fOptionsTIUTemplates changes 280](#_Toc475621396)

[6.2.2.2.28.3.4.1. Components on Form 280](#_Toc475621397)

[6.2.2.2.28.3.4.2. Events 280](#_Toc475621398)

[6.2.2.2.28.3.4.3. Class Methods 281](#_Toc475621399)

[6.2.2.2.28.3.4.4. Special References 281](#_Toc475621400)

[6.2.2.2.28.3.4.5. Class Events 281](#_Toc475621401)

[6.2.2.2.28.3.4.6. Class Properties 281](#_Toc475621402)

[6.2.2.2.28.3.4.7. Unit Variables 281](#_Toc475621403)

[6.2.2.2.28.3.4.8. Unit Methods 282](#_Toc475621404)

[6.2.2.2.28.3.5. File uTemplateFields.pas changes 283](#_Toc475621405)

[6.2.2.2.28.3.5.1. Components on Form 284](#_Toc475621406)

[6.2.2.2.28.3.5.2. Events 284](#_Toc475621407)

[6.2.2.2.28.3.5.3. Special References 284](#_Toc475621408)

[6.2.2.2.28.3.5.4. Class Events 284](#_Toc475621409)

[6.2.2.2.28.3.5.5. Unit Methods 284](#_Toc475621410)

[6.2.2.2.28.3.5.6. Class Properties 284](#_Toc475621411)

[6.2.2.2.28.3.5.7. Uses Clause 284](#_Toc475621412)

[6.2.2.2.28.3.6. Changes to fTemplateDialog.pas 286](#_Toc475621413)

[6.2.2.2.28.3.6.1. Current Form 286](#_Toc475621414)

[6.2.2.2.28.3.6.2. Modified Form 287](#_Toc475621415)

[6.2.2.2.28.3.6.3. Components on Form 288](#_Toc475621416)

[6.2.2.2.28.3.6.4. Events 288](#_Toc475621417)

[6.2.2.2.28.3.6.5. Methods 288](#_Toc475621418)

[6.2.2.2.28.3.6.6. Special References 288](#_Toc475621419)

[6.2.2.2.28.3.6.7. Class Events 289](#_Toc475621420)

[6.2.2.2.28.3.6.8. Class Methods 289](#_Toc475621421)

[6.2.2.2.28.3.6.9. Class Properties 289](#_Toc475621422)

[6.2.2.2.28.3.6.10. Uses Clause 289](#_Toc475621423)

[6.2.2.2.28.3.6.11. New Code added 290](#_Toc475621424)

[6.2.2.2.28.3.7. File mRequiredFieldsNavigatorpas changes 298](#_Toc475621425)

[6.2.2.2.28.3.7.1. Components on Form 298](#_Toc475621426)

[6.2.2.2.28.3.7.2. Special References 298](#_Toc475621427)

[6.2.2.2.28.3.7.3. Class Events 299](#_Toc475621428)

[6.2.2.2.28.3.7.4. Class Methods 299](#_Toc475621429)

[6.2.2.2.28.3.7.5. Class Properties 299](#_Toc475621430)

[6.2.2.2.28.3.7.6. New Code added 300](#_Toc475621431)

[6.2.2.2.28.3.7.7. Unit Methods 310](#_Toc475621432)

[6.2.2.2.28.3.8. File uRequiredFieldsHighlighter.pas changes 311](#_Toc475621433)

[6.2.2.2.28.3.8.1. Class Methods 311](#_Toc475621434)

[6.2.2.2.28.3.8.2. Special References 312](#_Toc475621435)

[6.2.2.2.28.3.8.3. Class Events 312](#_Toc475621436)

[6.2.2.2.28.3.8.4. Unit Methods 312](#_Toc475621437)

[6.2.2.2.28.3.8.5. New Code added 313](#_Toc475621438)

[6.2.2.2.28.3.9. File rOptions.pas changes 321](#_Toc475621439)

[6.2.2.2.28.4. CPRS Notification Alert Processing (NSR#20081008) 322](#_Toc475621440)

[6.2.2.2.28.4.1. Summary of changes 322](#_Toc475621441)

[6.2.2.2.28.4.2. Files changes 322](#_Toc475621442)

[6.2.2.2.28.4.3. Changes by Units 323](#_Toc475621443)

[6.2.2.2.28.4.3.1. Unit fPTSel 323](#_Toc475621444)

[6.2.2.2.28.4.3.2. Unit fAlertsProcessed 328](#_Toc475621445)

[6.2.2.2.28.4.3.3. Unit fOptions 344](#_Toc475621446)

[6.2.2.2.28.4.3.4. Unit rOptions 347](#_Toc475621447)

[6.2.2.2.28.4.3.5. Unit uConst 347](#_Toc475621448)

[6.2.2.2.28.4.3.6. Unit fOptionsPocessedAlerts 348](#_Toc475621449)

[6.2.2.2.28.4.3.7. Unit fAlertRangeEdit 353](#_Toc475621450)

[6.2.2.2.28.5. Progress Notes Display Misleading Request # 20070817 356](#_Toc475621451)

[6.2.2.2.28.5.1. Summary of changes 356](#_Toc475621452)

[6.2.2.2.28.5.2. Files changes 357](#_Toc475621453)

[6.2.2.2.28.5.3. Changes by Units 357](#_Toc475621454)

[6.2.2.2.28.5.3.1. Unit uDocTree 357](#_Toc475621455)

[6.2.2.2.28.5.3.2. Unit rTIU 362](#_Toc475621456)

[6.2.2.2.28.5.3.3. Unit fNotes 363](#_Toc475621457)

[6.2.2.2.28.5.3.4. Unit dShared 374](#_Toc475621458)

[6.2.2.2.29. GUI Classes 376](#_Toc475621459)

[6.2.2.2.30. Current Form 376](#_Toc475621460)

[6.2.2.2.30.1. Adverse Reaction Reporting File Modification Request #20120404 377](#_Toc475621461)

[6.2.2.2.30.2. Drug-Allergy Order Check Enhancements and Improved Detail Request #20100825 378](#_Toc475621462)

[6.2.2.2.30.3. Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request #20071216 379](#_Toc475621463)

[6.2.2.2.30.4. CWAD Post Auto-Demotion Rules Request #20080307 380](#_Toc475621464)

[6.2.2.2.30.5. Prevent Confusion over CPRS Status Display of Orders and Available Actions Request #20111006 381](#_Toc475621465)

[6.2.2.2.30.6. CPRS Notification Alert Processing Improvement Request #20081008 382](#_Toc475621466)

[6.2.2.2.30.7. Identify Required Fields in TIU Note Templates Request #20100706 382](#_Toc475621467)

[6.2.2.2.30.8. First Dose – Enhancement within CPRS Medication Order Request #20070811 384](#_Toc475621468)

[6.2.2.2.30.9. Critical / High Order Check Display Request #20101203 385](#_Toc475621469)

[6.2.2.2.30.10. D/C Order by Adverse Reaction Request #20080226 386](#_Toc475621470)

[6.2.2.2.30.11. Changes to Allergy/Pharmacy Packages Request #20071211 387](#_Toc475621471)

[6.2.2.2.30.12. Order Flag Recommendations Request #20110719 388](#_Toc475621472)

[6.2.2.2.30.13. Progress Notes Display Misleading Request #200070817 389](#_Toc475621473)

[6.2.2.2.31. Modified Form 389](#_Toc475621474)

[6.2.2.2.31.1. D/C Order by Adverse Reaction Request #20080226 389](#_Toc475621475)

[6.2.2.2.32. Components on Form 390](#_Toc475621476)

[6.2.2.2.33. Events 390](#_Toc475621477)

[6.2.2.2.34. Methods 395](#_Toc475621478)

[6.2.2.2.35. Special References 395](#_Toc475621479)

[6.2.2.2.36. Class Events 395](#_Toc475621480)

[6.2.2.2.37. Class Methods 396](#_Toc475621481)

[6.2.2.2.38. Class Properties 396](#_Toc475621482)

[6.2.2.2.39. Uses Clause 396](#_Toc475621483)

[6.2.2.2.40. Forms 396](#_Toc475621484)

[6.2.2.2.41. Functions 397](#_Toc475621485)

[6.2.2.2.42. Functions 400](#_Toc475621486)

[6.2.2.2.43. Dialog 402](#_Toc475621487)

[6.2.2.2.44. Help Frame 402](#_Toc475621488)

[6.2.2.2.45. HL7 Application Parameter 402](#_Toc475621489)

[6.2.2.2.46. HL7 Logical Link 403](#_Toc475621490)

[6.2.2.2.47. COTS Interface 414](#_Toc475621491)

[6.3. Network Detailed Design 414](#_Toc475621492)

[6.4. Security and Privacy 415](#_Toc475621493)

[6.4.1. Security 415](#_Toc475621494)

[6.4.2. Privacy 415](#_Toc475621495)

[6.5. Service Oriented Architecture / ESS Detailed Design 416](#_Toc475621496)

[7. External System Interface Design 416](#_Toc475621497)

[7.1. Interface Architecture 416](#_Toc475621498)

[7.2. Interface Detailed Design 416](#_Toc475621499)

[8. Human-Machine Interface 418](#_Toc475621500)

[8.1. Interface Design Rules 418](#_Toc475621501)

[8.2. Inputs 418](#_Toc475621502)

[8.3. Outputs 418](#_Toc475621503)

[8.4. Navigation Hierarchy 418](#_Toc475621504)

[8.4.1. Screen [x.1] 418](#_Toc475621505)

[8.4.2. Screen [x.2] 418](#_Toc475621506)

[8.4.3. Screen [x.3] 418](#_Toc475621507)

[9. Attachment A – Approval Signatures 418](#_Toc475621508)

[A. Additional Information 420](#_Toc475621509)

[A.1. Identification of Technology and Standards 420](#_Toc475621510)

[A.2. Constraining Policies, Directives and Procedures 420](#_Toc475621511)

[A.3. Requirements Traceability Matrix 420](#_Toc475621512)

[A.4. Packaging and Installation 420](#_Toc475621513)

[A.5. Design Metrics 420](#_Toc475621514)

# Introduction

This document is the System Design Document (SDD) for the Computerized Patient Record System (CPRS) v32 development project. Section 1 contains the purpose, scope, and references for the project.

This is a living document and will continue to evolve throughout the project. This project is implemented utilizing agile methods which include multiple sprints, or short development cycles. This document will be updated regularly during the development cycle to reflect the changes implemented in a sprint along with changes planned for future sprints.

## Scope

Link to [CPRS v32 Consolidated BRD](http://URL)

## User Profiles

There are several types of users who utilize CPRS. Primarily, the intended user are clinicians such as physicians, nurses and pharmacists who are dealing with patient care. In addition, there are laboratorians, financial staff and others who use CPRS to find patient-related information. These users are trained on how to utilize and customize it to fit their needs per their required functionality. There are a large number of documents and presentations on many of functionalities of CPRS, including online training. While proficiency levels vary among the different users, there is support provided via the context-sensitive help, various CPRS support teams, web sites and training on new CPRS features.

The Information Technology (IT) staff that support CPRS are typically supporting multiple systems and are familiar with support and maintenance of VistA systems. Documentation is provided with each release of CPRS on proper installation techniques and support is provided by the development team during deployment. In addition, there are staff available to assist the IT representatives should problems arise.

# Background

## Overview of the System

Computerized Patient Record System (CPRS) provides an integrated patient record system for clinicians, managers, Quality Assurance (QA) staff, and researchers. The primary goal of CPRS is to provide a fast and easy-to-use application that makes available to providers the information needed in the clinical workflow process. CPRS user interface is integrated with Veterans Health Information Systems and Technology Architecture (VistA) to facilitate reviewing, documenting and preserving of coordinated care information and improved accessibility of online clinical information and results.

The CPRS v32 project consists of over 28 New Service Requests (NSRs). Some of the more significant NSRs are:

* Drug Allergy Order Check (20100825)
* Changes to Nurse Order Verification (20090416)
* Add Address of Performing Lab to Reports (20081206)
* Prevent Confusion over CPRS Status Display (20111006)
* Change in Unflagging Capabilities (20071103)
* CPRS Notification Alert Processing (20081008)
* Identify Req Flds TIU Note Template Req (20100706)
* Update Surrogate Management Functionality (20071216)
* Adverse Reaction Reporting File Mods (20120404)
* Allergy Order Check Enhancement (20070203)
* Button Link No Assessment Warn to Allergy (20070920)
* Real Time Notification of Pot. Missed Orders (20060710)
* Enhanced Allergy Checks –COTS (20080704)
* Changes to Allergy/Pharmacy Packages (20071211)
* Order Flag Recommendations (20110719)

At a high level, CPRS v32’s modifications focus on improving patient safety and the display and handling of clinical data in CPRS and several ancillary VistA packages. Specifically, some of the critical CPRS v32 requests contain improvements in the following areas:

* Allergy order checking – multiple modifications surrounding handling of allergy information; including improvements in patient care and safety
* Improved display of clinical data, such as IV medication administration, for the clinical users of CPRS.
* Improved patient instructions on both medication labels and discharge summaries in the area of Outpatient prescriptions.
* Increased patient safety related to renewal of Outpatient prescriptions.
* Software to support business process changes related to processing Inpatient Medications and making them available to BCMA for administration.
* Reduce confusion over displays in CPRS related to Orders and Available actions, as well as selection of provider names when names are similar.
* Workflow and safety improvements in the area of flagging medication orders.

Subject Matter Experts (SMEs) from affected areas such as physcians, nurses and pharmacists as well as clinical application coordinators have been involved in determining the final requirements and will be involved in the field testing and approval of these enhancements for release.

## Overview of the Business Process

CPRS v32 supports the following business processes:

* Order flagging/unflagging
* Allergy order checks
* Critical order check overrides
* No Allergy Assessment handling
* TIU Templates
* CPRS Chart Display
* Lab Reporting in CPRS
* Nurse Order Verification
* Immunizations

Link to [CPRS v32 RSD](http://URL/projects/CPRS/v32/TeamDocs/HP%20General/Archive%20-%20Old%20Documents/CPRSv32%20RSD%20v1.14%20(November%202015)_Archive.docx)

## Overview of the Significant Requirements

There is a link to the consolidated RSD in section 2.2.

Link to [Consolidated BRD](http://URL/warboard/ProjectDocs/CPRS_version_32/CPRS%20Consolidated%20NSR%20BRD.pdf)

Rational Overview Document. We are currently in the process of converting to Rational. This will be linked as soon as that is complete.

High-lever requirements description:

### Drug-Allergy Order Check Enhancements and Improved Detail Request (#20100825) (RSD #2.6.1)

The Drug Allergy Order Check enhancement provides the ability for the user to be notified of possible adverse drug reactions when orders are placed through either CPRS or the IP Medications applications. IP Medications currently only references the first ingredient in the “Previous Adverse Reaction” display, which poses a problem if a reactant contains multiple ingredients.

### Changes to the Nurse Order Verification’s Effect on Order Status Request (#20090416) (RSD #2.6.2)

This enhancement improves patient safety by preventing medication errors; these errors occur when orders that the pharmacist has not completed reviewing are inadvertently activated by the nurse’s verification.

### Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216) (RSD #2.6.3)

This is an enhancement to the surrogate setting screen within the CPRS GUI. This screen allows for a user to “forward” patient-related notifications to another user during absences from the work place. Currently the CPRS GUI only allows a single surrogate to be displayed, however the List Manager interface to Kernel allows for a single or multiple/sequential surrogate to be set and displayed back to the user for confirmation. This enhancement shall provide the same functionality to the CPRS GUI as is currently available via the List Manager/Kernel surrogate management settings, so that the same functionality is provided regardless of interface used, and the same actions can be undertaken in CPRS GUI within normal provider workflows

### CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4)

This is an enhancement to Crisis Notes, Warning Notes, Allergies and Directives (CWAD) of CPRS. CWAD is a section of CPRS used for posting progress notes, which are more important than standard level notes. These progress notes are made more easily available throughout CPRS. The postings box can become full of notes, resulting in important notes to be easily non-distinguishable. The enhancement is to demote the note from postings to regular status based on various criteria, such as the passage of time or a newer note of a particular title has been written superseding the existing one.

### Add Address of Performing Lab to Reports in CPRS Health Summary Request (#20081206) (RSD #2.6.6)

This enhancement shall (a) have the facility address display in the header of all Health Summary Reports and (b) add the “performing” lab’s name and address to all lab reports printed from the Health Summary within CPRS. The performing lab is the lab (could be a non-Department of Veterans Affairs [VA] lab) where the test was processed. The enhancement shall support maintaining the old address on historical lab reports and showing a new address on lab report if the facility moves.

### CPRS Day of the Week Med Schedule Change Request (#20120802) (RSD #2.6.7)

This enhancement changes CPRS canned Inpatient Medication Schedule of "Other or day of the week" to expand the abbreviations with the full day of the week (i.e., Thursday, not Th) when an Inpatient order is copied to an Outpatient order by the provider.

Currently, patients may receive printed discharge instructions and possibly outpatient prescription labels with unclear directions such as Th@1700. Expanding the abbreviation to Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday would clear up any confusion caused by the abbreviation.

### Prevent Confusion over CPRS Status Display of Orders and Available Actions Request (#20111006) (RSD #2.6.8)

This enhancement addresses the confusion over the status of display of orders and actions available. Items for review include inconsistent use of the terms "DISCONTINUE/CANCEL", ramifications of unsigned orders and disparate functionality between the MEDS tab and ORDERS tab, among others.

### Change in Unflagging Capabilities Request (#20071103) (RSD #2.6.9)

This is an enhancement to the CPRS GUI that restricts the ability to perform the unflag function on any order. Order flagging is one way to send a notification seeking additional information or clarification about an order to another user. Currently, any user can remove a flag associated with an order in CPRS that is intended for a specific recipient. This creates a situation where the original recipient will not be notified of the flag; this can create delays in patient care and potential patient safety issues.

### CPRS Notification Alert Processing Improvement Request (#20081008) (RSD #2.6.10)

This is an enhancement to Notification Alerts in the CPRS GUI. In CPRS, alerts (also referred to as notifications), are specific to the user signed into the system and are displayed to the user on the Patient Selection screen. Alerts can be processed one at a time, or all can be processed by stepping through each alert using a “Next” button. However, if during alert processing the alert being processed is an “information only” alert, the alert is not retained beyond the initial display to the user. If the user steps through the alerts too quickly, there is no means within CPRS to redisplay the information alert.

### Confirm Provider Selected with Similar Names Request (#20110606) (RSD #2.6.11)

This is an enhancement to CPRS involving the development of an additional selection window that could be presented within CPRS for providers with similar names, similar to what is currently done with patients who have the same last name and last four (4) digits of the Social Security Number (SSN). Currently, providers with the same last name, i.e., HILL,Mxx RMH 11A PHYSICIAN/HILL,Rxx RCH 112 PHYSICIAN, are often selected incorrectly. If users are not careful when indicating which provider is responsible for an order, or with the assignment of additional signers in notes, the wrong provider is selected.

### Identify Required Fields in TIU Note Templates Request (#20100706) (RSD #2.6.12)

This is an enhancement to CPRS that identifies what data is missing in a required field within the CPRS Text Integration Utility (TIU). Dialog style progress note templates can contain dialog fields that are required to be completed. When a required field is missed, the user is not currently directed to the specific required field that was missed. Instead, the user is now only prompted with "one or more required fields must be entered". As the user is not given direction as to which required field has been missed, finding the missing required field is often very difficult.

### Adverse Reaction Reporting File Modification Request (#20120404) (RSD #2.6.13)

This is an enhancement to the Adverse Reaction Reporting File. Currently for adverse reactions, the severity rating is not required and all historical entries must have a SIGN/SYMPTOM or COMMENT of the reaction included. This leads to confusion by staff when reviewing past reactions regarding proper medication selection. This enhancement is required to improve documentation of observed historical adverse reactions to include the following:

* Require selection of at least one sign/symptom (currently this is not a mandated field).
* Add UNKNOWN as a sign/symptom choice for historical adverse reactions only (send a request to New Term Rapid Turnaround).
* When an UNKNOWN sign or symptom is selected, require documentation of a COMMENT.
* Add the Severity field to the dialogue so that it can be documented, if it is known.

### Allergy Order Check Enhancement Request (#20070203) (RSD #2.6.14)

This is an an enhancement that checks for interactions against existing medications when a new allergy is entered. Patient allergies are documented directly using the Adverse Reaction Tracking (ART) application or indirectly through CPRS. Documentation of allergies is done by qualified individuals, such as the attending physician, clinical pharmacist, or clinical pharmacologist. In CPRS, allergies can be entered on the cover sheet tab or the orders tab. Allergies are verified by qualified individuals. When an allergy or adverse reaction has been entered into the system, if the new order includes a medication containing that reactant, the system displays an alert to the provider. However, when an allergy is entered into the record of a patient that has active orders, the system does not have the ability to check the active drug profile to determine if there is a potential drug-allergy interaction with the active orders. NSR 20070203 addresses this shortcoming by scanning the patient’s active profile for medications containing the new reactant and sending Nofifications to those recipients defined in CPRS parameters. A form is also displayed allowing the user to optionally select additional recipients to Notify.

### Park-A-Prescription (#20090509) (RSD #2.6.15)

This is an enhancement to address concerns with the ‘hold’ status for outpatient prescriptions and make changes to Pharmacy medication reconciliation.

The Park-A-Prescription initiative began as a Veterans Health Administration Innovation Program (VHAIP) and went through the Innovations Development Cycle. The Park-A- Prescription codebase was developed to be implemented into CPRS version 28 using Delphi 2006. The need is to integrate this to a nationally exportable and functional code under CPRS version 32, which is based off of Delphi XE3. The majority of this work is estimated to be refactoring the Park-A-Prescription codebase from Delphi 2006 to Delphi XE3 and addressing any issues that come about from refactoring.

### First Dose – Enhancement within CPRS Medication Order Request (#20070811) (RSD #2.6.16)

This is an enhancement to automate the steps for monitoring the patient’s response to the first dose(s) of a medication new to the patient under the direct care of the hospital. Joint Commission Health Administration Services (HAS) requires "First Dose" observation and documentation of a patient’s response to the first dose(s) of a medication new to them when they are under the direct care of the hospital. This functionality does not currently exist

### IV Administration Not Documented in CPRS Request (#20110903) (RSD #2.6.17)

This is an enhancement to address consistent infusion times between Bar Code Medication Administration (BCMA) and CPRS. A problem was reported with documentation of administration times of intravenous (IV) electrolyte boluses in the medications tab of CPRS. A patient had two separate orders for a Sodium Phosphate 10mmol bolus administered to the patient on two separate days. However, both showed as administered on one date in the CPRS meds tab as CPRS only shows completion time.

### Critical / High Order Check Display Request (#20101203) (RSD #2.6.18)

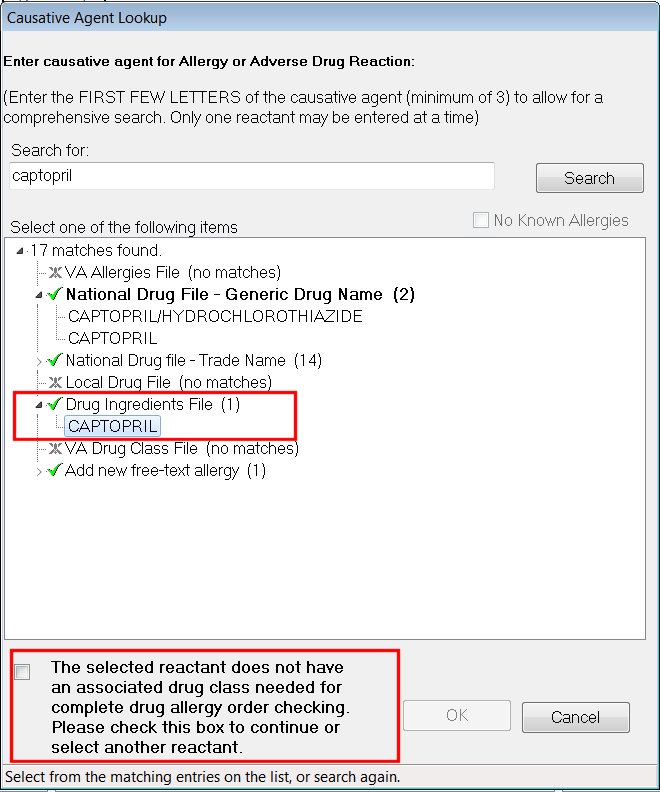
This is an enhancement to the existing CPRS order check functionality. Currently, there is only one prompt presented to clinicians to justify all CPRS order checks for a patient. In addition, regular/low-risk order checks are grouped together with critical/high-risk order checks in the same window. This causes confusion among clinicians about which order checks actually require justification and have higher priority over others. To remove this confusion, individual display windows for each degree of order checks need to be established with individual justification prompts for each order check that requires a physician override. This would force the provider to be aware of all order checks, with overlapping priority given to critical/high-risk order checks, and would require the provider to independently enter a justification for each order check before the order would be allowed to become active.

### Button to Link No Assessment Warning to Allergy Assessment Screens Request (#20070920) (RSD #2.6.19)

This is an enhancement that would require clinicians ordering medications in CPRS, when presented with the warning “No allergy assessment has been done,” to complete the Allergy Assessment on patients for whom there is no such information, before they can proceed to the medication orders screen.

### Real-time Notification of Potentially Missed Order Checks Request(#20060710) (RSD #2.6.20)

This enhancement will notify a provider in real-time that Drug Class cross checking will not occur based on their selection of the Drug Ingredient option (which does not perform Drug Class cross checking). The notification includes a warning and a checkbox that must be checked in order to proceed thereby overriding the warning (See screenshot below).



### Enhanced Allergy Checks Using COTS Data Request (#20080704) (RSD #2.6.21)

This is an enhancement that will provide the ability to update and maintain chemical drug classification system to the current Adverse Reaction Tracking (ART), or allergy system, to ensure the allergy documentation system can provide alerts for allergies to drugs of a similar chemical structure. This will lead to fewer adverse reactions and will increase usefulness of the documented allergy information.

Please see the MOCHA Server SDD for information about modifications to MOCHA Server.

### D/C Order by Adverse Reaction Request (#20080226) (RSD #2.6.22)

This is an enhancement that will provide the ability for a provider to optionally enter an Adverse Drug Reaction (ADR) if discontinuing a medication for reason of an allergy to the medication.

### Changes to Allergy/Pharmacy Packages Request (#20071211) (RSD #2.6.23)

These are several enhancements related to allergy notifications and allergy alerts triggered during the medication order process that will work toward eliminating the potential for patient allergy information being overlooked by clinical staff

### Order Flag Recommendations Request (#20110719) (RSD #2.6.24)

This is an enhancement to address potential hazards with the use of Flag Order functions that have been identified by the CPRS Clinical Workgroup. A CPRS Clinical Workgroup subgroup documented potential hazards with the use of Order Flags, from the perspectives of (a) the user who sets the flag, (b) the user who processes the Flagged Order Alert, and (c) the user who unflags the order. Patient care can be compromised due to these hazards which related to several IT Patient Safety issues.

### Progress Notes Display Misleading Request (#20070817) (RSD #2.6.25)

This is an enhancement to the Text Integrated Utility (TIU) Progress Notes Display in CPRS. Several VA facilities have reported that providers are missing information found on progress notes that they do not see displayed, and this has been classified as a Patient Safety issue.

### Indication for Prescription and Medication Orders (#20100101) (RSD #2.6.26)

This is an enhancement to provide the ability to enter the Indication for Use for prescriptions and medications that are ordered. The system will provide a method for providers to associate a medication indication with each medication order they place in CPRS, Inpatient Medications or IV Medications and Outpatient Pharmacy.

# Conceptual Design

## Conceptual Application Design

**Drug-Allergy Order Check Enhancements and Improved Detail Request (#20100825)**

Link to [Drug Allergy Order Check Enhancements and Improved Detail BRD](http://URL/pasdocs/analysis/20100825_Drug-Allergy_Order_Check_Enhancements_and_Improved_Detail_BRD.doc)

**As-Is:**



**To-Be:**



**Changes to the Nurse Order Verification’s Effect on Order Status Request (#20090416)**

Link to [Changes to the Nurse Orde Verification's Effect Order Status BRDh](http://URL/pasdocs/analysis/20090416%20Changes%20to%20the%20Nurse%20Order%20Verification's%20Effect%20on%20Order%20Status%20BRD.doc)

**As-Is:**



**To-Be:**



**Add Address of Performing Lab to Reports in CPRS Health Summary Request (#20081206)**

Link to [Add Address of Performing Lab to Reports in CPRS Health Summary BRD](http://URL/pasdocs/analysis/20081206%20Add%20Address%20of%20Performing%20Lab%20to%20Reports%20in%20CPRS%20Health%20Summary%20BRD.docx)

**As-Is/To-Be:**



**Prevent Confusion Over CPRS Status Display of Orders and Available Actions Request (#20111006)**

Link to [Prevent Confusion Over CPRS Status Display Screening Call Record](http://URL/pasdocs/ScreeningCallRecord/20111006%20Prevent%20Confusion%20Over%20CPRS%20Status%20Display_Screening_Call_Record.doc)

**As-Is:**





**To-Be:**

**Change in Unflagging Capabilities Request (#20071103)**

Link to [Change in Unflagging Capabilities BRD](http://URL/pasdocs/analysis/Change%20in%20Unflagging%20Capabilities%20BRD%2020071103.doc)

As-Is:

Change in Unflagging Capabilities Request (#20071103). Original Workflow

**To-Be**



**CPRS Notification Alert Processing Improvement Request (#20081008)**

Link to [CPRS Alert Processing Notification Improvement BRD](file:///C:\Users\PII)

Identify Required Fields in TIU Note Templates Request (#20100706)

Data is available at, no UML documents, but business process flow remains the same with identifying information added to required fields.

Link to [Identify Required Fields in TIU Note Templates Requesth](http://URL/nsrd/Tab_LinksView.asp?RequestID=20100706)

Adverse Reaction Reporting File Modification Request (#20120404)

As is Business Flow Diagram available, process remains the same with an additional field marked as required.

Link to [Adverse Reaction Reporting Supporting Docsh](http://URL/pasdocs/supportinfo/20120404%20Adverse%20Reaction%20Reporting%20Supporting%20Docs.zip)

Allergy Order Check Enhancement Request (#20070203)

Link to [Allergy Order Check Enhancement BRD](http://URL/pasdocs/analysis/20070203%20Allergy%20Order%20Check%20Enhancement%20BRD.doc)

First Dose – Enhancement within CPRS Medication Order Request (#20070811)

Link to [First Dose Enhancement within CPRS Med Orders BRD](http://URL/pasdocs/analysis/20070811%20First%20Dose%20Enhancement%20within%20CPRS%20Med%20Orders%20BRD.doc)

Button to Link No Assessment Warning to Allergy Assessment Screens Request (#20070920)

Supporting NSR documentation (no business process diagrams.)

Link to [Tab LinksView Request 20070920](http://URL/nsrd/Tab_LinksView.asp?RequestID=20070920)

Real-time Notification of Potentially Missed Order Checks Request(#20060710)

Link to [Real-Time Notification of Potentially Missed Order Checks BRD](http://URL/pasdocs/analysis/Real-Time%20Notification%20of%20Potentially%20Missed%20Order%20Checks%20%20BRD%2020060710.doc)

Enhanced Allergy Checks Using COTS Data Request (#20080704)

Link to [Enhanced Allergy Checks Using COTS Data](http://URL/pasdocs/analysis/20080704%20Enhanced%20Allergy%20Checks%20Using%20COTS%20Data.zip)

Changes to Allergy/Pharmacy Packages Request (#20071211)

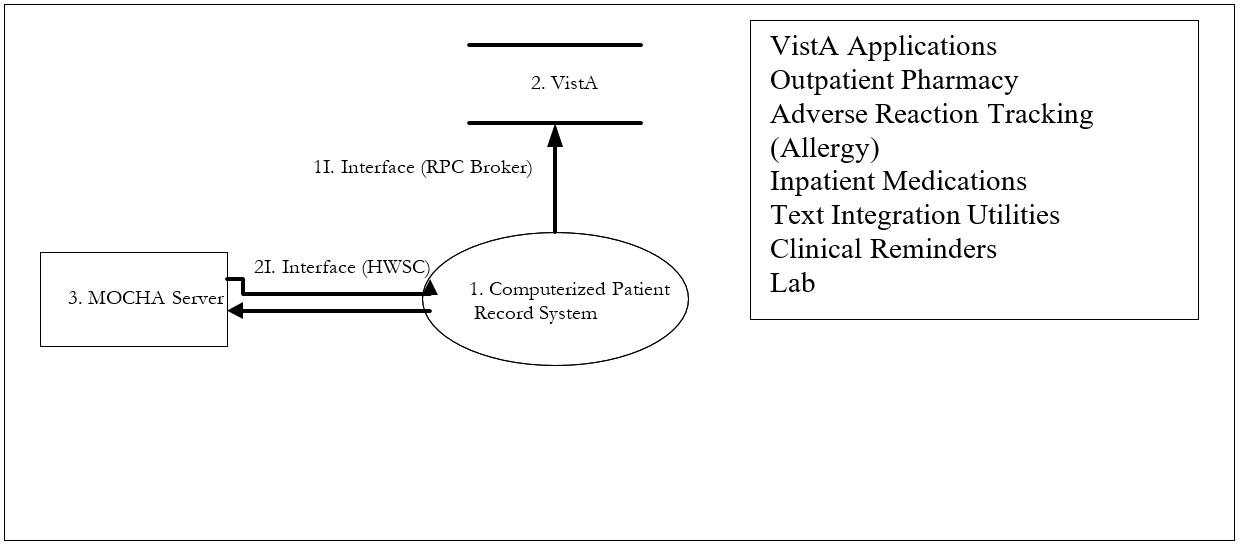
Link to [Changes to Allergy Pharmacy Packages BRD](http://URL/pasdocs/analysis/20071211%20Changes%20to%20Allergy-Pharmacy%20Packages%20BRD%20.zip)

Order Flag Recommendations Request (#20110719)

Link to [Order Flag Recommendations Request](http://URL/nsrd/Tab_LinksView.asp?RequestID=20110719)

### Application Context

There is no modification to the existing application context. Please refer to the MOCHA Server SDD for changes to the MOCHA Server interface.



Application Context Diagram

Table 5 (Grouping): Application Context Description

Object

| ID | Name | Description | Interface Name | Interface System |
| --- | --- | --- | --- | --- |
| 1 | Computerized Patient Record System (CPRS) | CPRS (1) retrieves and sends clinical information to VistA (2). This includes order information such as Pharmacy Prescriptions. | RPC Broker | VistA |
| 2 | VistA | VistA is the primary data store for Patient and Health related information. | RPC Broker | CPRS |
| 3 | MOCHA Server | MOCHA Server contains information from First Data Bank (FDB) such as drug-drug interaction, or allergy data | HealtheVet  Web Services Client (HWSC) | VistA |

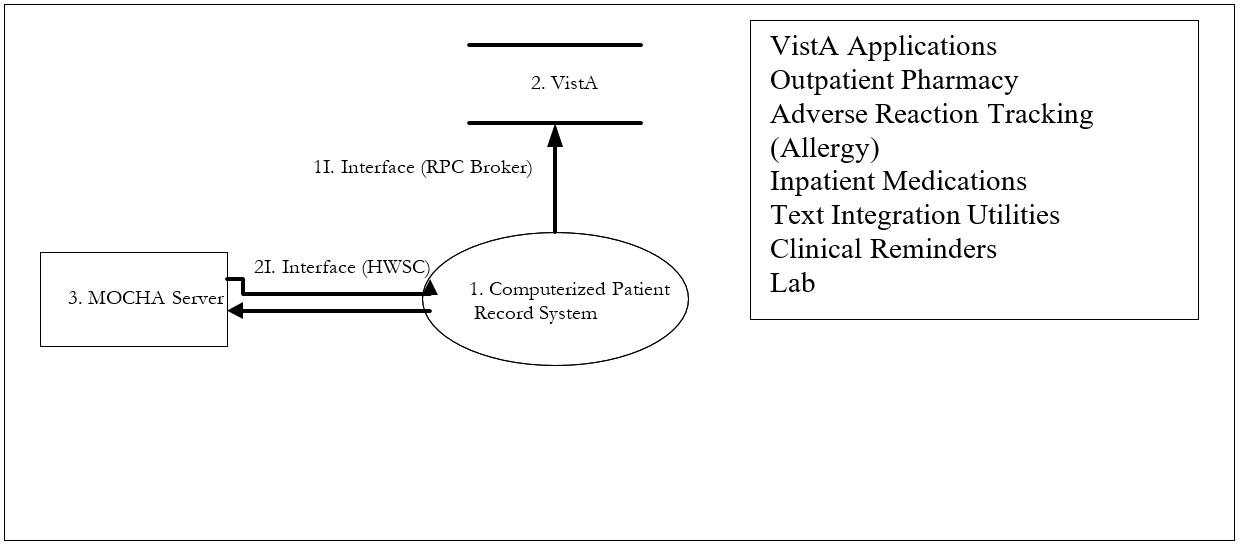
Interfaces External to OIT

CPRS does not contain any interfaces external to OIT.

Interfaces Internal to OIT

| ID | Name | Related Object | Input Messages | Output Messages | External Party |
| --- | --- | --- | --- | --- | --- |
| 1I | RPC Broker | CPRS (1)  VistA (2) | Message containing a request for information or order information | Requested information, or notification of success. | N/A |
| 2I | HWSC | VistA (2)  MOCHA Server (3) | Message containing a request for information or order information | Drug interactions, Allergy Data | N/A |

### High-Level Application Design



High-Level Application Design

Table 6: Objects in the High Level Application Design

Objects / Components to be Built or Modified

Internal Data Stores

| ID | Name | Data Stored | Steward | Access |
| --- | --- | --- | --- | --- |
| 2 VistA (Adverse Reaction Tracking/ Allergy) | Adverse Reaction Tracking | Chemistry based allergy information. | GMRA – Adverse Reaction Tracking | Create  Retrieve  Update  Delete |

### Application Locations

CPRS is a Legacy VistA application. Therefore, even though Cloud technology is a goal in reforming Federal IT, this project will not utilize that technology. In addition, regional deployments are used at various stations, however, it is not the decision of the development team. Regional deployment decisions are made by the regional IT staff.

Table 7: Application Locations

| Application Component | Description | Location at Which Component is Run | Type |
| --- | --- | --- | --- |
| CPRS | Executable | Workstations located throughout each VAMC. | Presentation Logic although there is some business logic contained in the executable. |
| VistA Server | Database which also contains the server side executable code of the VistA applications. | VistA servers are located either in local VAMCs or at regional data processing centers. | There are some applications that use the roll/scroll presentation logic. It is also the database and contains business logic and interface code. |

Table 8: Application Users

| Application Component | Location | User |
| --- | --- | --- |
| CPRS and VistA Server | Users for both CPRS and the VistA server are located not only locally at each VAMC, but can also be located at CBOCS or may be located at remote location and connect through CAG or VPN | Physician  Nurse  Pharmacist  Laboratorian  While the previous list shows the main users affected by changes introduced with this project, essentially, anyone involved in patient care will use CPRS and any user of VistA will be accessing the VistA server |

## Conceptual Data Design

### Project Conceptual Data Model

Project Conceptual Data Model



### Database Information

Table 9: Database Inventory

| Database Name | Description | Type | Steward |
| --- | --- | --- | --- |
| VistA | M/Cache database containing clinical and non-clinical data for all VistA application | Modify | Overall, the VA is the steward of the database in its’ entirety.  Specifically, data is being modified in data that is owned by Pharmacy, CPRS, Kernel and Laboratory |

### User Interface Data Mapping

#### Application Screen Interface

##### Discontinue / Cancel Orders

**Error! Reference source not found.**) represents the screen that displays the orders to be cancelled and the list of predefind reasons that the clinicianmust select one reason before advancing to the ADR detail screen. Table 10 describes it.

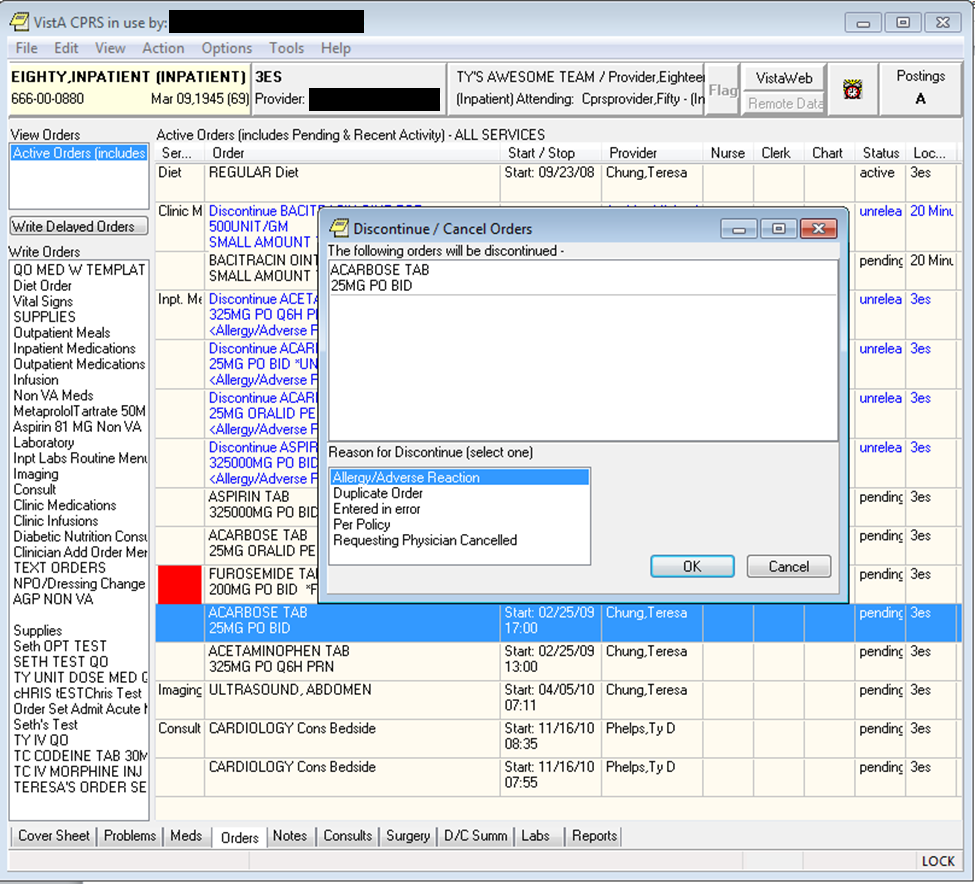


Figure 5: Discontinue / Cancel Orders Screen

##### List of allergies currently recorded for the patient

**Error! Reference source not found.**” represents the screen that displays known Allergies previously discontued for current patient. If the medication is on the list the cliention, should select the No button to cancel the addition of the allergy. If the Alergy is not on the list, the cliention will select the Yes button to add a Discontued/ Adverse reaction.

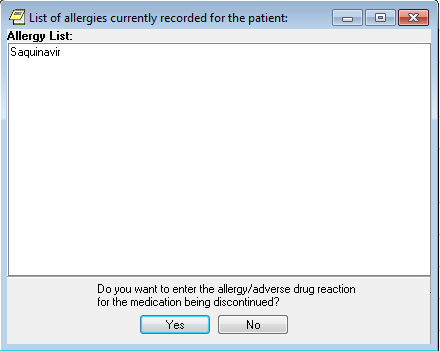
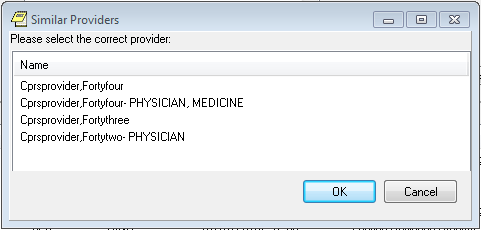


Figure: List of allergies currently recorded for the patient Screen

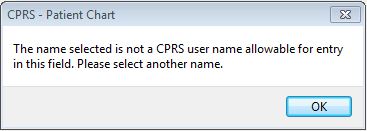
##### Similar Provider Screen

The Similar Provider screen shows the listing of names that match the desired pattern of the lastname and first two(2) letters of the firstname for a provider that was selected and where similar names have been found. From this screen the proper provider name will be chosen. It is actually just some addition coding that was implemented in the fDupPts.pas/dfm unit rather than creating a separate form for Similar Providers.

Figure: Similar Patient, Provider and CosignerScreen and Message for Non-Qualified Selection.

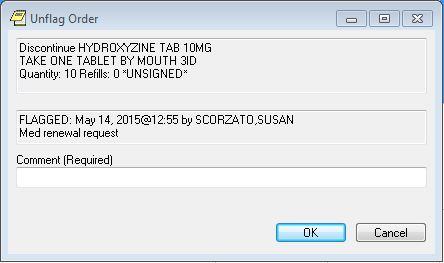


**Figure: Similar Provider Message**



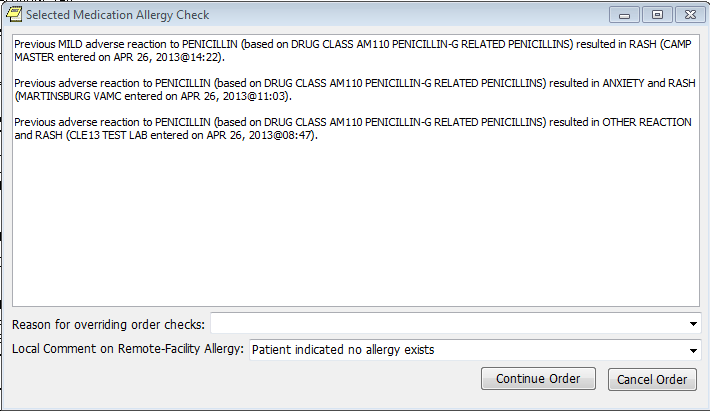
##### Unflagging Orders Screen

**Error! Reference source not found.**The main change to this screen is that there is now a Required comment to be entered whenever an order has been unflagging and notification sent to the selected list of recipients.



##### Allergy Pharmacy Packages Check Screen

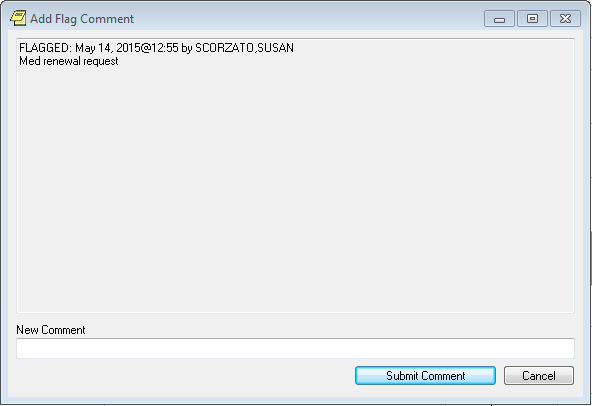
The Allergy Pharmacy Check screen display override reasons assigned to a specific medication for a specific patient and also allows for a new override reason to be assigned and displaying a dropdown listing of pre-defined reasons that can be chosen from as well as enter free text for any reason deemed applicable to this issue. There is also a field for a comment to be added to this check for any remote reason that has been identified.



Allergy Pharmacy Check Screen

##### Order Flagging Screens

Figure: 10: First screen allows for a comment to be required when an order is flagged, second screen is to allow for a selection of any number of selected recipients to be chosen for notification of order flagging.



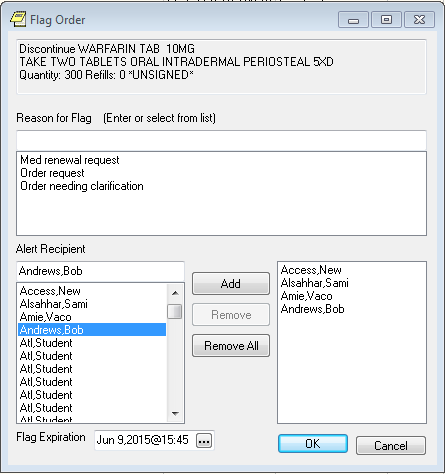


Figure 11: Order Flagging Recommendations Screen

Table 10: : Discontinue / Cancel Orders Screen Description

| Graphical User Interface (GUI) Field | Table (Database Table that field connects to) | Field (Field in Table that the GUI field connects to) | Comments |
| --- | --- | --- | --- |
| lstOrders | Patient file |  |  |
| lstReason | Field - DCReason | RPC - 'ORWDX2 DCREASON' | The list of DC Reasons is sorted ascending during the RPC execution on the VISTA server |

#### Application Report Interface

##### Add Performing Lab to Report

There are multiple reports affected by this modification. They are currently under development. Screen shots will be added as they are completed.

Table 11: *<Report name>* Description

| Report Column | Data Source *<Table Name. Fieldname>* |
| --- | --- |
| Patient | <xxx.PATIENT\_NAME> |
| SSN | <xxx.SSN> |
| DoB | <yyyy.DATE\_OF\_BIRTH> |

#### Unmapped Data Element

Currently, there are no new data elements being added that are not mapped to a user interface.

## Conceptual Infrastructure Design



### System Criticality and High Availability

CPRS is considered a mission critical system. However, without the VistA database and servers, CPRS cannot function. OIT has designed regional data processing centers and ‘hot backup’ sites for VistA instances. This project is not modifying those approaches. Any modification to those schemes will be done outside the consideration of these enhancements. Special Technology

### Special Technology

N/A

### Technology Locations

Table 13: Technology Location Details

| Technology Component  Production 1 | Location | Usage |
| --- | --- | --- |
| Workstations | VAMCs  CBOCs  Remote (through CAG and VPN) | Clinicians and support staff |
| Special Hardware | None |  |
| Interface Processors | Local to VAMCs  Austin AITC  Regional | Interfaces to First DataBank via MOCHA server  Interface to HDR |
| Legacy Mainframe | None |  |
| Legacy Application Server | None |  |
| Legacy Databases | Local  Regional | Houses VistA database. |
| Other | None |  |

### Conceptual Infrastructure Diagram

#### Location of Environments and External Interfaces

Currently, the location of environments varies between VAMCs. Some VAMCs still have local hardware configurations. Many are converted to a regional data processing setup. There is a group responsible for determining the appropriate configuration and scheduling conversions for sites from local to regional deployment.

CPRS v32 is designed to fit within whatever environment exists: either local or regional.

#### Conceptual Production String Diagram



Conceptual Production String Diagram

# System Architecture

## Hardware Architecture

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. As such, it utilizes the existing VistA hardware architecture.

Some VistA instances are local to the VAMC that use it. Others are installed in regional data processing centers. Most of these instances have hot backup sites located in another geographically separated location.

The primary architecture at this time is a cluster of Linux servers that act as the applications server. The client workstations are connected to this cluster. This cluster is connected to a cluster of VMS servers that act as the database server where the VistA database resides.

The client workstations may be local to the VAMC or they may be remote at CBOCs, other VAMCs (in the case of integrated sites) or may even be one-off remote workstations that connect via CAG or VPN.

While personnel outside of Product Development are responsible for determining the best configurations and ensuring adequate hardware and network connectivity, the CPRS v32 project considers additional space and potential performance impacts. The ultimate goal is to add no more than an additional 5% disk space or CPU requirements.

During the field testing phase, any additional files created are monitored to ensure this is not exceeded. Local and regional IT staff monitor the systems and notifie CPRS development is the performance exceeds expectations based on the development environment.



## Software Architecture

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is not changing that underlying software architecture.

CPRS v32 utilizes Delphi and Mumps as the primary programming languages. The only exception to that are minor changes being made to the MOCHA server that will be done in Java.

CPRS v32 is currently using Perforce as the source code control system. The team is in the process of converting to Rational as the source code control system.

CPRS v32 requires no additional software to support the system – outside of the existing VistA infrastructure.

## Network Architecture

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. The executable portion of CPRS v32 will continue communicate using the existing network architecture that supports the legacy VistA systems.

CPRS v32 utilizes remote procedure calls (RPCs) over the local or wide area network to communicate between the client and the VistA instance. This communication utilizes Kernel’s broker package.

Reference the hardware architecture for a high-level overview of the communication pathways.

## Service Oriented Architecture / ESS

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is enhancing the existing CPRS system.

Refer to the [VistA Monograph](http://www.ehealth.va.gov/VistA_Monograph.asp) for a full explanation of the larger system that CPRS v32 is a part of.

Note: CPRS’s architecture does not supply new services or consume services.

## Enterprise Architecture

CPRS v32 uses Delphi XE3 for the Delphi development. Currently, we are using XE3 which is one of the Divest status TRM items. The plan is to move away from XE3 on a future version.

The server side code is written using Cache/MUMPS, which is approved under the TRM.

# Data Design

## DBMS Files

CPRS v32 will utilize the existing FileMan database. There are some minor updates being made to the existing PRESCRIPTION file (#52) which contains dynamic data that will be updated whenever the Park a Prescription functionality is utilized.

In addition, it is anticipated that one requirement will add a new static file. This particular modification is still being designed. When that design is complete, the details will be added. It is anticipated that this file will be built upon installation and rarely, if ever, updated. The file is also expected to contain fewer than 100 entries and will be accessed whenever a critical or allergy order check requires an override reason. CPRS v32 is currently gathering statistics on the frequency of these types of order checks.

Section 6.2.2 contains the specifics of the data dictionary updates.

The files will be accessed either by FileMan Database Server (DBS) calls or by direct global access, where appropriate and allowed.

## Non-DBMS Files

N/A

## Data View

A "Data View" should be included in the Architectural Representation whenever persistent data objects are included in the system (they are typically present in most software systems). The data view describes the logical data model of the system and includes an Entity Relationship Diagram (ERD). For a description of Entity Relationship diagramming please refer to the whitepaper <<http://www-106.ibm.com/developerworks/rational/library/content/03July/2500/2785/2785_uml.pdf>>

There are no updates planned for the persistent data objects currently in the released version of CPRS. Since no Entity Relationship Diagram (ERD) can be located CPRS v32 will construct a high level ERD to be included in the final versio of this document.

# Detailed Design

## Hardware Detailed Design

N/A. CPRS v32 will use existing hardware infrastructure and will be designed to require no significant increase in data storage capacity or CPRS resources.

## Software Detailed Design

### Conceptual Design

#### Product Perspective

View CPRS v32 requirements in this link: [Computerized Patient Record System (CPRS) v32 Requirements Specification Document](http://URL/warboard/ProjectDocs/CPRS_version_32/CPRSv32%20RSD%20v1.5%20(Feb%202015).docx)

##### User Interfaces

The following three block diagrams show, at its simplest form the interfaces in use for CPRS v32. There will be two GUI executables: CPRS and BCMA. Both of these utilize GUI screens to communicate with users.

For all VistA (server side) changes, the user interface will be roll and scroll.



##### Hardware Interfaces

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is enhancing the existing CPRS system. No modifications to the existing hardware interfaces are planned.

Currently, CPRS, utilizing the Kernel system will support any device that Kernel supports. CPRS v32 does not control this support.

##### Software Interfaces

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is enhancing the existing CPRS system.

CPRS communicates with all the VistA clinical applications and several of the financial applications. In addition, CPRS uses FileMan and Kernel. CPRS is written using the current nationally released version of each of these packages and plans to continue to utilize and support the currently released version.

##### Communications Interfaces

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is enhancing the existing CPRS system. No modifications will be made to the existing communications interface which are under the control of Engineering.

##### Memory Constraints

N/A

##### Special Operations

N/A

#### Product Features

Please refer to section 2.5.

#### User Characteristics

CPRS is used primarily by clinicians such as physicians, nurses, physician assistants, nurse practioners, pharmacists and other ancillary clinical users.

#### Dependencies and Constraints

CPRS v32 must utilize the existing hardware and network infrastructure. Therefore, the increase in network, memory, CPU and data storage can not be significant. Some remote installations, such as the CBOCs are particularly susceptible.

CPRS v32 must be 508 compliant.

The design must take into account that CPRS is a mission critical application. As an example: if access to First DataBank is currently off line, CPRS must continue to function.

### Specific Requirements

#### Database Repository

The data for CPRS v32 will reside in the existing VistA database.

The following new fields under the History multiple (#999.1) of the Institution file (#4) were created for **Add Address of Performing Lab to Reports in CPRS Health Summary (#20081206)**:

4.999,1 STREET ADDR. 1 (CHANGED FROM) 1;1 FREE TEXT

INPUT TRANSFORM: K:$L(X)>40!($L(X)<2) X

LAST EDITED: JUN 10, 2016

HELP-PROMPT: Answer must be 2-40 characters in length.

DESCRIPTION: If STREET ADDR. 1 changed, this field holds old value.

4.999,1.1 STREET ADDR. 2 (CHANGED FROM) 1;2 FREE TEXT

INPUT TRANSFORM: K:$L(X)>40!($L(X)<2) X

LAST EDITED: JUN 10, 2016

HELP-PROMPT: Answer must be 2-40 characters in length.

DESCRIPTION: If STREET ADDR. 2 changed, this field holds old value.

4.999,1.2 CITY (CHANGED FROM) 1;3 FREE TEXT

INPUT TRANSFORM: K:$L(X)>40!($L(X)<2) X

LAST EDITED: JUN 10, 2016

HELP-PROMPT: Answer must be 2-40 characters in length.

DESCRIPTION: If CITY changed, this field holds old value.

4.999,1.3 STATE (CHANGED FROM) 1;4 POINTER TO STATE FILE (#5)

LAST EDITED: JUN 10, 2016

HELP-PROMPT: Enter the state.

DESCRIPTION: If STATE changed, this field holds old value.

4.999,1.4 ZIP (CHANGED FROM) 1;5 FREE TEXT

INPUT TRANSFORM: K:$L(X)>10!($L(X)<5) X

LAST EDITED: JUN 10, 2016

HELP-PROMPT: Enter ZIP code or ZIP + 4, 5-10 characters in length.

DESCRIPTION: If ZIP changed, this field holds old value.

The following new Security Key multiple (#100.985) was added to the Display Group file (#100.98) for **Change in Unflagging Capabilities (#20071103)**:

100.98,5 SECURITY KEY 2;0 POINTER Multiple #100.985

DESCRIPTION: A list of Security Keys assigned to a Display

Group to be used for validating a user's

privileges when attempting to unflag an order.

100.985,.01 SECURITY KEY 0;1 POINTER TO SECURITY KEY FILE (#19.1)

LAST EDITED: NOV 10, 2016

HELP-PROMPT: Enter a Security Key from file 19.1 to be

assigned to this Display Group

DESCRIPTION: A Security Key assigned to this Display Group

to be used for validating a user's privileges

when attempting to unflag an order.

CROSS-REFERENCE: 100.985^B

1)= S ^ORD(100.98,DA(1),2,"B",$E(X,1,30),DA)=""

2)= K ^ORD(100.98,DA(1),2,"B",$E(X,1,30),DA)

The following new fields in the Patient Allergies (#120.8) file were created for **Adverse Reaction Reporting File Modifications (#20120404)**:

120.8,8       HISTORICAL SEVERITY    0;8 SET

                                '1' FOR MILD;

                                '2' FOR MODERATE;

                                '3' FOR SEVERE;

              LAST EDITED:      SEP 22, 2016

              HELP-PROMPT:      Enter the severity of the historical

                                allergy/adverse reaction.

              DESCRIPTION:      The severity of a historical allergy/adverse

                                reaction.

120.8,9       HISTORICAL DATE/TIME OF EVENT 0;9 DATE

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

              LAST EDITED:      DEC 19, 2016

              HELP-PROMPT:      Enter the date/time (in the past) this

                                historical allergy/adverse reaction occurred.

              DESCRIPTION:      The date/time a historical allergy/adverse

                                reaction occurred.

The following new cross-reference in the Alert Tracking (#8992.1) file was created for **Notification Alert Processing Improvement (#20081008)**:

Subfile #8992.11

Record Indexes:

PAR (#1564) RECORD REGULAR IR LOOKUP & SORTING

WHOLE FILE (#8992.1)

Short Descr: Indexed by Processed Alert & Recipient

Description: Cross-reference on entire file that first sorts on

Processed Alert (date field) and then on Recipient (pointer

to New Person file)

Set Logic: S ^XTV(8992.1,"PAR",X(1),X(2),DA(1),DA)=""

Kill Logic: K ^XTV(8992.1,"PAR",X(1),X(2),DA(1),DA)

Whole Kill: K ^XTV(8992.1,"PAR")

X(1): PROCESSED ALERT (8992.11,.04) (Subscr 1) (forwards)

X(2): RECIPIENT (8992.11,.01) (Subscr 2) (forwards)

#### System Features

Please refer to the RSD using this link:

[Computerized Patient Record System (CPRS) v32 Requirements Specification Document](http://URL/warboard/ProjectDocs/CPRS_version_32/CPRSv32%20RSD%20v1.5%20(Feb%202015).docx)

##### NSR # 20070203 - Allergy Order Check Enhancement

###### ORWDAL32

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDAL32 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call | ORWPS, ORWDXC, XPAR, XQALERT |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Added Function CHKMEDS (Called by RPC ORWDAL32 CHKMEDS) with subfunction CHKMEDS2:**  CHKMEDS(LST,ORDFN,GMRAGNT) ;Check a newly entered allergy against existing orders  N ALST,L,MED,M,AGYLST,ORD,ENT,DFN,ATTEND,MDA,MEDD,MDARRAY,MDARRAY2  S LST=0  D ACTIVE^ORWPS(.ALST,ORDFN,DUZ,0,0)  S L="" K ORD F S L=$O(ALST(L)) Q:L="" I $E(ALST(L))="~" D  . S MED=$P(ALST(L),U,9),MEDD=$P(ALST(L),"^",3) I $D(^OR(100,+MED,.1)) D  . . S MDA=0 F S MDA=$O(^OR(100,+MED,.1,MDA)) Q:MDA=""!(MDA'?1N.N) I $  D(^OR(100,+MED,.1,MDA,0)) D  . . . S M=^OR(100,+MED,.1,MDA,0),MDARRAY(M,+MED)=MEDD  I $D(MDARRAY) D  . S M="" F S M=$O(MDARRAY(M)) Q:M="" I $D(MDARRAY(M)) D  . . K AGYLST  . . D ALLERGY^ORWDXC(.AGYLST,ORDFN,"PSI",M,M)  . . I $$CHKMEDS2($P(GMRAGNT,U),.AGYLST) D  . . . S MED="" F S MED=$O(MDARRAY(M,MED)) Q:MED="" S MDARRAY2(MED,M)=  MDARRAY(M,MED)  . K MDARRAY  I $D(MDARRAY2) D  . S MED="" F S MED=$O(MDARRAY2(MED)) Q:MED="" D  . . S ORD=$P($G(^OR(100,MED,0)),U,4),ENT=$P($G(^OR(100,MED,0)),U,6)  . . S M="" F S M=$O(MDARRAY2(MED,M)) Q:M="" D  . . . S LST=LST+1  . . . S LST(LST)=MED\_U\_M\_U\_MDARRAY2(MED,M)  . . . I ORD]"" S $P(LST(LST),U,4)=ORD\_";"\_$P(^VA(200,ORD,0),U,1)  . . . I ENT]"",ORD'=ENT S $P(LST(LST),U,5)=ENT\_";"\_$P(^VA(200,ENT,0),U,1)  . . . S DFN=$P($G(^OR(100,+MED,0)),U,2),ATTEND=$G(@(U\_$P(DFN,";",2)\_$P(  DFN,";")\_",.1041)"))  . . . I ATTEND]"",ORD'=ATTEND S $P(LST(LST),U,6)=ATTEND\_";"\_$P(^VA(200,  ATTEND,0),U,1)  . K MDARRAY2  K ALST,AGYLST  Q  CHKMEDS2(AGNT,AGYLST) ;Scan returned allegy checks against the new allergy agent  for a match  N MATCH,AGY  S MATCH=0,AGY=""  F S AGY=$O(AGYLST(AGY)) Q:AGY="" I AGYLST(AGY)[AGNT S MATCH=1  Q MATCH |
| **Added Function GETPROV (Called by RPC ORWDAL32 GETPROV):**  GETPROV(LST,ORDLST) ;return a list of providers related to a list of orders based on parameter option  N INST,PARM,ENT,ORDFN,PLST,DFN,PROV,TEAM,PTYP,PID,PSTR,PNAM,ATTEND,ORD  K PLST  S ENT=XPARSYS\_"^170;DIC(9.4,"  S INST=$O(^ORD(100.9,"B","ALLERGY ADDED ON ACTIVE MED",""))  ;I INST="" S LST=0 Q ;Quit if the alert has not been defined yet  S LST=0  S PARM=$$GET^XPAR(ENT,"ORB PROVIDER RECIPIENTS",INST,"Q")  I INST="" S PARM="PTAOE" ;Default to option "PTAOE"  S ORDFN=ORDLST  D  . S DFN=$P($G(^OR(100,+ORDFN,0)),U,2),ORD=$P($G(^OR(100,+ORDFN,0)),U,4),ENT=$P($G(^OR(100,+ORDFN,0)),U,6)  . S DFN=U\_$P(DFN,";",2)\_$P(DFN,";")\_","  . I PARM["P" S PROV=$P($G(@(DFN\_"""PC"""\_")")),U) I PROV]"" S PLST("PROV",PROV)="" ;Primary Provider  . I PARM["T" S TEAM=$P($G(@(DFN\_"""PC"""\_")")),U,2) I TEAM]"" S PLST("TEAM",TEAM)="" ;Patient Care Team  . I PARM["A" S ATTEND=$G(@(DFN\_".1041)")) I ATTEND]"" S PLST("PROV",ATTEND)="" ;Attending Provider  . I PARM["O",ORD]"" S PLST("PROV",ORD)="" ;Ordering Provider  . I PARM["E",ENT]"" S PLST("PROV",ENT)="" ;Entering User  ;. I PARM["M" PCMM Team  ;. I PARM["R" PCMM Primary Care Practitioner  ;. I PARM["S" PCMM Associate Provider  ;. I PARM["C" PCMM Mental Health Treatment Coordinator  F PTYP="PROV","TEAM" D  . S PID="" F S PID=$O(PLST(PTYP,PID)) Q:PID="" D  . . S LST=LST+1  . . S PSTR=PID\_";"\_$S(PTYP="PROV":"VA(200,",1:"SCTM(404.51,")  . . S PNAM=$P($G(@(U\_$P(PSTR,";",2)\_$P(PSTR,";")\_",0)")),U,1)  . . S LST(LST)=PSTR\_U\_PNAM  Q |
| **Added Function SENDALRT (Called by RPC ORWDAL32 SENDALRT):**  SENDALRT(Y,ORDLST,PROVLST) ;Send a group of alerts for instances where a user enters a new allergy impacting an existing med order  ;ORDLST contains a list of orders for which the alert will be sent  ; Format: Order Id^Medication Id^Medication Name^Ordered By^Entered By^Attending Provider  ;PROVLST contains a list of additional recipients selected by the user  ; Format: Entity Pointer^Entity Name  ; Entity can be a user {VA(200,)} or a group {SCTM(404.51}  N ORN,ORBN,ORD,ORD1,PROV,PROV1,PXQA,ORBDFN  S Y="",ORD=0,PROV=0  S ORN=$O(^ORD(100.9,"B","ALLERGY ADDED ON ACTIVE MED",""))  I ORN="" S Y="0^Alert not yet set-up on this system" Q  S ORBN=^ORD(100.9,ORN,0)  N XQA,XQAARCH,XQACNDEL,XQADATA,XQAFLG,XQAGUID,XQAID,XQAMSG,XQAOPT,XQAREVUE,XQAROU,XQASUPV,XQASURO,XQATEXT  F S PROV=$O(PROVLST(PROV)) Q:PROV="" D  . I PROVLST(PROV)["SCTM(404.51" S PXQA("G.MAILGROUP",$P(PROVLST,";"))="" Q  . S PXQA($P(PROVLST,";"))=""  F S ORD=$O(ORDLST(ORD)) Q:ORD="" D  . K XQA M XQA=PXQA  . F ORD1=4:1:6 I $P(ORDLST(ORD),U,ORD1)]"" S XQA($P($P(ORDLST(ORD),U,ORD1),";"))=""  . S ORBDFN=$P($G(^OR(100,+ORDFN,0)),U,2)  . S XQAID=$P(ORBN,"^",2)\_","\_+ORBDFN\_","\_ORN  . S XQAMSG="Review New Allergy Entered on Active Med ("\_$P(ORDLST,U,3)\_")",XQADATA=$P(ORDLST,U,1,3)  . I $D(XQA) D SETUP^XQALERT  S Y=1  Q |

###### ORDV06D

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORDV06D | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20070203 (**Allergy Order Check Enhancement**) | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| No current logic for new functionality |

| Modified Logic |
| --- |
| Created new routine ORDV06D based on ORDV06B for the new Active Medications with Allergies report. IN^ORDV06D is the extraction routine and RXACT^ORDV06D is the report routine.  ORDV06D ; SLC/JAS - OE/RR Report Extracts ;07/28/16 13:36  ;;3.0;ORDER ENTRY RESULTS REPORTING;\*\*405\*\*;Dec 17, 1997;Build 1  ;Pharmacy Extracts for CPRS Active Meds w/ Allergies Report  ;Copied from ORDV06B and modified to specific report needs  RXACT(ROOT,ORALPHA,OROMEGA,ORMAX,ORDBEG,ORDEND,OREXT) ;All Active Patient Meds  ;Call to PSOORRL  I $L($T(GCPR^OMGCOAS1)) D ; Call if FHIE station 200  . D GCPR^OMGCOAS1(DFN,"RXOP",ORDBEG,ORDEND,9999)  ;  N ORRXSTAT,GO  Q:'$L(OREXT)  S GO=$P(OREXT,";")\_"^"\_$P(OREXT,";",2)  Q:'$L($T(@GO))  S ORRXSTAT=""  D GETMED  Q  IN ;Setup and call to Pharmacy API  ;LST(i)=  ;LST(i) flags: "~" Start of new record, "/" Continuation line (concatination with Line feed CRLF)  ;  ;{ 1 2 3 4 5 6 7 8 9 10 11 16  ;{ Pieces: Typ^PharmID^Drug^InfRate^StopDt^RefRem^TotDose^UnitDose^OrderID^Status^LastFill^...^StartDt^ }  ;If $P($P(X,"^",2),";",2)= "I" or "C" then Inpatient=TRUE  ;If $P(X,"^",1)="~NV" then NonVAMed=TRUE and Instruct="Non-VA "\_Instruct  ;If $E($P(X,"^",1),1,2)="t\" then this is a comment, strip off the 1st character (t) and concatenate to other text  ;Location := $P($P(X,U,1),":",2);  K ^TMP("PS",$J),^TMP("ORACT",$J),^TMP("ORPS",$J)  N ORBEG,OREND,ERROR,ORCTX,ORVIEW  S (ORBEG,OREND,ORCTX)=""  S ORVIEW=3 I $G(ORDEND)="" S ORDEND=DT  S ORBEG=$S($G(ORDBEG):ORDBEG,1:$$DT^ORWPS("T-50000")),OREND=$S(ORDEND<DT:ORDEND,1:$$DT^ORWPS("T+3000"))  D OCL^PSOORRL(DFN,$$DT^ORWPS("T-50000"),$$DT^ORWPS("T+3000"),ORVIEW)  N ITMP,FIELDS,INSTRUCT,COMMENTS,REASON,NVSDT,TYPE,ILST,J,SORTDT,STOPDT  S ILST=0,ITMP=""  F S ITMP=$O(^TMP("PS",$J,ITMP)) Q:'ITMP D  . K INSTRUCT,COMMENTS,REASON,ORIFN  . K ^TMP("ORACT",$J,"COMMENTS")  . S COMMENTS="^TMP(""ORACT"",$J,""COMMENTS"")"  . S (INSTRUCT,@COMMENTS,STOPDT)="",FIELDS=^TMP("PS",$J,ITMP,0)  . I $P(FIELDS,"^",9)["DISCONTINUED"!($P(FIELDS,"^",9)["EXPIRED")!($P(FIELDS,"^",9)["CANCELLED") Q  . S $P(FIELDS,"^",17)=$P($G(^TMP("PS",$J,ITMP,"P",0)),"^",2) ;Provider  . S SORTDT=$S($L($P(FIELDS,"^",10)):$P(FIELDS,"^",10),1:$P(FIELDS,"^",15)) ;Date Priority: 1)Last Fill Date, 2)Issue/Start Date, 3)Order Date  . I 'SORTDT D ;If pharmacy API doesn't screen out data within selected date range, check CPRS OrderDate and screen out as appropriate  .. K ^TMP("ORXPS",$J) M ^TMP("ORXPS",$J)=^TMP("PS",$J)  .. D OEL^PSOORRL(DFN,$P(FIELDS,"^")) ;This API uses same ^TMP("PS" global  .. S ORIFN=+$P(^TMP("PS",$J,0),"^",11) I ORIFN S SORTDT=$P(^OR(100,ORIFN,0),"^",7),STOPDT=$P(^(0),"^",9)  .. M ^TMP("PS",$J)=^TMP("ORXPS",$J) K ^TMP("ORXPS",$J)  . S TYPE=$S($P($P(FIELDS,U),";",2)="O":"OP",1:"UD")  . I $D(^TMP("PS",$J,ITMP,"CLINIC",0)) S TYPE="CP"  . N LOC,LOCEX S (LOC,LOCEX)=""  . I TYPE="CP" S LOC=$G(^TMP("PS",$J,ITMP,"CLINIC",0))  . S:LOC LOCEX=$P($G(^SC(+LOC,0)),U)\_":"\_+LOC ;IMO  . I TYPE="OP",$P(FIELDS,";")["N" S TYPE="NV" ;non-VA med  . ;Next line excludes any data where (ExpirationDate, LastFill Date, StartDate or OrderDate) is outside of selected date range for everything except non-VAmeds.  . I TYPE'="NV",SORTDT<ORBEG!(SORTDT>OREND),($P(FIELDS,"^",4)<ORBEG!($P(  FIELDS,"^",4)>OREND)),($P(FIELDS,"^",10)<ORBEG!($P(FIELDS,"^",10)>OREND)),($P(FIELDS,"^",15)<ORBEG!($P(FIELDS,"^",15)>OREND)) Q  . I $P(FIELDS,"^",9)["DISCONTINUED",(TYPE="OP"!(TYPE="NV")) D  .. K ^TMP("ORXPS",$J) M ^TMP("ORXPS",$J)=^TMP("PS",$J)  .. D OEL^PSOORRL(DFN,$P(FIELDS,"^")) ;This API uses same ^TMP("PS" global  .. S ORIFN=+$P(^TMP("PS",$J,0),"^",11) I ORIFN S STOPDT=$P(^OR(100,ORIFN,0),"^",9)  .. M ^TMP("PS",$J)=^TMP("ORXPS",$J) K ^TMP("ORXPS",$J)  .. I TYPE="NV",'$L($P(FIELDS,"^",4)) S $P(FIELDS,"^",4)=STOPDT  .. I TYPE="OP" S $P(FIELDS,"^",4)=STOPDT  . I $O(^TMP("PS",$J,ITMP,"A",0))>0 S TYPE="IV"  . I $O(^TMP("PS",$J,ITMP,"B",0))>0 S TYPE="IV"  . I (TYPE="UD")!(TYPE="CP") D UDINST^ORWPS(.INSTRUCT,ITMP)  . I TYPE="OP" D OPINST^ORWPS(.INSTRUCT,ITMP)  . I TYPE="IV" D IVINST^ORWPS(.INSTRUCT,ITMP) |

. I TYPE="NV" D NVINST^ORWPS(.INSTRUCT,ITMP),NVREASON^ORWPS(.REASON,.NVSDT,ITMP)

. I (TYPE="UD")!(TYPE="IV")!(TYPE="NV")!(TYPE="CP") D SETMULT^ORWPS(COMMENTS,ITMP,"SIO")

. M COMMENTS=@COMMENTS

. I $D(COMMENTS(1)) S COMMENTS(1)="\"\_COMMENTS(1)

. I '$L($P(FIELDS,U,15)) S:TYPE="NV" $P(FIELDS,U,15)=$P($G(NVSDT),".")

;Set Start Date for non-VA Med (from file 100, which currently doesn't get set)

. I LOC S ^TMP("ORPS",$J,$$NXT)="~CP:"\_LOCEX\_U\_FIELDS

. E S ^TMP("ORPS",$J,$$NXT)="~"\_TYPE\_U\_FIELDS

. S J=0 F S J=$O(INSTRUCT(J)) Q:'J S ^TMP("ORPS",$J,$$NXT)=INSTRUCT(J)

. S J=0 F S J=$O(COMMENTS(J)) Q:'J S ^TMP("ORPS",$J,$$NXT)="t"\_COMMENTS(J)

. S J=0 F S J=$O(REASON(J)) Q:'J S ^TMP("ORPS",$J,$$NXT)="t"\_REASON(J)

K ^TMP("PS",$J),^TMP("ORACT",$J)

Q

NXT() ; increment ILST

S ILST=ILST+1

Q ILST

;

GETMED ;

N J,ORIPS,ORIPSS,ORRXNO,ORSTAT,ORQTY,OREXP,ORISSUE,ORLAST,ORREF,ORPRVD,

ORCOST,ORSIG,ORT,ORX0

N ECD,GMR,GMW,IX,PSOBEGIN,GMTSNDM,GMTS1,GMTS2,ORSITE,SITE,X,NONVA,INST,

OLDORI,RT,X,X2,X3,ORII,ORKK

N AFND,ORORD,ILST

S ORSITE=$$SITE^VASITE,ORSITE=$P(ORSITE,"^",2)\_";"\_$P(ORSITE,"^",3)

;Sorted by STATUS then by DRUG NAME

K ^TMP("ORDATA",$J),^TMP("ORT",$J)

I '$L($T(GCPR^OMGCOAS1)) D

. K ^TMP("ORPS",$J)

. D @GO

S (OLDORI,ORIPS,ORT)=0

F S ORIPS=$O(^TMP("ORPS",$J,ORIPS)) Q:(ORIPS'>0) S X=$G(^(ORIPS)) I X'="" D

. I $E(X)="~" D Q

.. S OLDORI=ORIPS,ORT=0,X3=$S($L($P(X,"^",10)):$P(X,"^",10),1:"ZUNKNOWN"),X2=$S($L($P(X,"^",3)):$P(X,"^",3),1:"ZUNKNOWN")

.. S ^TMP("ORT",$J,X3,X2,ORIPS)=X

. I $L(X2),$L(X3),$E(X)="\" S ORT=ORT+1,^TMP("ORT",$J,X3,X2,OLDORI,ORT)=$E(X,2,9999)

S ORII=""

F S ORII=$O(^TMP("ORT",$J,ORII)) Q:ORII="" S ORKK="" F S ORKK=$O(^TMP("ORT",$J,ORII,ORKK)) Q:ORKK="" D

. S ORIPS=0 F S ORIPS=$O(^TMP("ORT",$J,ORII,ORKK,ORIPS)) Q:(ORIPS'>0) S ORX0=^(ORIPS),AFND=0 D

.. I $E(ORX0)="~" D Q

... S ORIPSS=$S($L($P(ORX0,U,10)):$E($P(ORX0,U,10),1,10),1:"UNK")\_"\_"\_$S($L($P(ORX0,U,3)):$P(ORX0,U,3),1:"UNK")\_"\_"\_ORIPS

... S ORORD=$P($P(ORX0,U,9),";") D ACHK Q:'AFND

... S ^TMP("ORDATA",$J,ORIPSS,"WP",1)="1^"\_ORSITE ;Station ID

... S ^TMP("ORDATA",$J,ORIPSS,"WP",2)="2^"\_$P(ORX0,U,3) ;Medication Name

... S ^TMP("ORDATA",$J,ORIPSS,"WP",3)="3^"\_$P(ORX0,U,10) ;Status

... S X=$P($P(ORX0,"^",2),";",2),^TMP("ORDATA",$J,ORIPSS,"WP",4)="4^"\_$S(X="I":"IN",X="C":"IN",1:"OUT") ;In/OutPatient

... S X=$P(ORX0,"^"),^TMP("ORDATA",$J,ORIPSS,"WP",5)="5^"\_$S(X="~NV":"NonVAMed",1:"RX") ;Type: RX or NonVA Med

... S ^TMP("ORDATA",$J,ORIPSS,"WP",6)="6^"\_$$DATE^ORDVU($P(ORX0,U,16)) ;Start Date

... S ^TMP("ORDATA",$J,ORIPSS,"WP",7)="7^"\_$$DATE^ORDVU($P(ORX0,U,5)) ;Stop Date

... S ^TMP("ORDATA",$J,ORIPSS,"WP",8)="8^"\_$$DATE^ORDVU($P(ORX0,U,11)) ;Last Fill Date

... S ^TMP("ORDATA",$J,ORIPSS,"WP",9)="9^"\_$P(ORX0,U,18) ;Provider

... S ^TMP("ORDATA",$J,ORIPSS,"WP",12)="12^[+]" ;flag for detail

... S ORT=0 F S ORT=$O(^TMP("ORT",$J,ORII,ORKK,ORIPS,ORT)) Q:'ORT S X=^(ORT),^TMP("ORDATA",$J,ORIPSS,"WP",10,ORT)="10^"\_X ;Instructions

K ^TMP("ORPS",$J),^TMP("ORXPND",$J),^TMP("ORT",$J)

S ROOT=$NA(^TMP("ORDATA",$J))

Q

ACHK ;

N ORIDA,ORORI,ORPXI,ORPXI2,ORDRGDA,ORDCDA,ILST

; Order check for drug allergies

Q:'$D(^OR(100,ORORD,.1))

S ILST=1,ORIDA=0

F S ORIDA=$O(^OR(100,ORORD,.1,ORIDA)) Q:ORIDA="" I $D(^OR(100,ORORD,.1,ORIDA,0)) D

. S ORORI=^OR(100,ORORD,.1,ORIDA,0)

. Q:'$D(^ORD(101.43,ORORI,0))

. S ORPXI=$P($P(^ORD(101.43,ORORI,0),"^",2),";"),ORPXI2=$P($P(^ORD(101.43,ORORI,0),"^",2),";",2)

. K ^TMP($J,"ORDRGLST")

. D DRGIEN^PSS50P7(ORPXI,"","ORDRGLST") ; ICR 4662

. Q:'^TMP($J,"ORDRGLST",0)

. S ORDRGDA=0 F S ORDRGDA=$O(^TMP($J,"ORDRGLST",ORDRGDA)) Q:ORDRGDA="" D

. . K ^TMP($J,"ORDRGCL"),ORDRGCL

. . D ZERO^PSS50(ORDRGDA,"","","","","ORDRGCL") ; ICR 4483

. . Q:$G(^TMP($J,"ORDRGCL",ORDRGDA,2))=""

. . S ORDRGCL(^TMP($J,"ORDRGCL",ORDRGDA,2))=ORDRGDA

. Q:'$D(ORDRGCL)

. S ORDCDA="" F S ORDCDA=$O(ORDRGCL(ORDCDA)) Q:ORDCDA="" D

. . K ORALL

. . S ORDRGDA=ORDRGCL(ORDCDA)

. . S AFND=$$ORCHK2^GMRAOR(DFN,"DR",".."\_ORDRGDA,"","ORALL") ; ICR 2378

. . I AFND D ADETAIL

. K ^TMP($J,"ORDRGLST"),^TMP($J,"ORDRGCL"),ORDRGCL

Q

ADETAIL ;

; Set detail lines from ORALL array

N ADA,ASDA,ASEVDA

S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^"

F ADA=1:1:ORALL I $D(ORALL(ADA,"MESSAGE")) D

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Causative agent: "\_$P(ORALL(ADA,"MESSAGE",2),"^",2)

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^"

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Symptons: "\_$P(ORALL(ADA,"MESSAGE",2),"^",1)

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Drug Class: "\_$G(ORALL(ADA,"MESSAGE","OFFENDERS","CLS"))

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^"

. S ASDA="" F S ASDA=$O(ORALL(ADA,"MESSAGE",1,ASDA)) Q:ASDA="" I $D(ORALL(ADA,"MESSAGE",1,ASDA)) D

. . S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Originator: "\_$P(ORALL(ADA,"MESSAGE",1,ASDA),"^",1)

. . S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Originated: "\_$P(ORALL(ADA,"MESSAGE",1,ASDA),"^",3)

. . S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^Observed/Historical: "\_$P(ORALL(ADA,"MESSAGE",1,ASDA),"^",4)

. . I $D(ORALL(ADA,"MESSAGE",1,ASDA,1)) S ASEVDA="" F S ASEVDA=$O(ORALL(ADA,"MESSAGE",1,ASDA,1,ASEVDA)) Q:ASEVDA="" I $D(ORALL(ADA,"MESSAGE",1,ASDA,1,ASEVDA)) D

. . . S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^ Obs Dates/Severity: "\_$P(ORALL(ADA,"MESSAGE",1,ASDA,1,ASEVDA),"^",1)\_" - "\_$P(ORALL(ADA,"MESSAGE",1,ASDA,1,ASEVDA),"^",2)

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^"

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^------------------------

-------------------------------"

. S ^TMP("ORDATA",$J,ORIPSS,"WP",11,$$NXT)="11^"

Q

###### GMRAPEM0

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAPEM0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20070203 (**Allergy Order Check Enhancement**) | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| SELECT ;Select a patient reaction  S GMRACNT=0 D 1^VADPT  S GMRALOC=$P(VAIN(4),U,2),GMRANAM=VADM(1),GMRASEX=VADM(5),GMRAOUT=0,GMR  AOTH=$O(^GMRD(120.83,"B","OTHER REACTION",0)) D KVAR^VADPT K VA,VAROOT  K GMRADUP S GMRALAGO=1  D REACT^GMRAPAT(DFN) ; Load all reaction for this patient.  D EN1^GMRAPES0  I GMRAPA>0 D TYPE D  .I GMRAOUT D:$G(GMRANEW) DELETE S:'$$MISSREQ&('$P($G(GMRAPA(0)),U,12))  GMRAOUT=0,^TMP($J,"GMRASF","B",GMRAPA,GMRACNT)="",^TMP($J,"GMRASF",GMRACNT,GMRAP  A)="" D:GMRAOUT UPOUT^GMRAPEM3 Q ; 21,36  .I GMRAERR D ERR^GMRAPEM3 Q ;The reaction was entered in error  .I $P(GMRAPA(0),U,12) D SIGNED^GMRAPEM3 Q ;The reaction has been signe  d  .; Reaction is a new reaction or Update data  .I GMRANEW D GMRACHK^GMRAPEM0(GMRAPA)  .D UPDATE^GMRAPEM3  .Q  Q |

| Modified Logic (Changes are in bold) |
| --- |
| SELECT ;Select a patient reaction  S GMRACNT=0 D 1^VADPT  S GMRALOC=$P(VAIN(4),U,2),GMRANAM=VADM(1),GMRASEX=VADM(5),GMRAOUT=0,GMR  AOTH=$O(^GMRD(120.83,"B","OTHER REACTION",0)) D KVAR^VADPT K VA,VAROOT  K GMRADUP S GMRALAGO=1  D REACT^GMRAPAT(DFN) ; Load all reaction for this patient.  D EN1^GMRAPES0  I GMRAPA>0 D TYPE D  .I GMRAOUT D:$G(GMRANEW) DELETE S:'$$MISSREQ&('$P($G(GMRAPA(0)),U,12))  GMRAOUT=0,^TMP($J,"GMRASF","B",GMRAPA,GMRACNT)="",^TMP($J,"GMRASF",GMRACNT,GMRAP  A)="" D:GMRAOUT UPOUT^GMRAPEM3 Q ; 21,36  .I GMRAERR D ERR^GMRAPEM3 Q ;The reaction was entered in error  .I $P(GMRAPA(0),U,12) D SIGNED^GMRAPEM3 Q ;The reaction has been signe  d  .; Reaction is a new reaction or Update data  .I GMRANEW D GMRACHK^GMRAPEM0(GMRAPA)  **.I GMRANEW D MEDCHK ; NSR 20070203**  .D UPDATE^GMRAPEM3  .Q  Q  **New Code:**  **MEDCHK ; NSR 20070203**  **N GMRAMCHK**  **W !!,?2,"Checking new allergy against the patient's active medication profile . . . "**  **D CHKMEDS^ORWDAL32(.GMRAMCHK,$P(GMRAPA(0),"^"),$P(GMRAPA(0),"^",2)) ; Need ICR**  **I GMRAMCHK'>0 W !!,?2,"No conflicts were discovered.",! K GMRAMCHK Q**  **I GMRAMCHK>0 D**  **. N GMRADA,GMRAYN,GMRAPROV,GMRASEND,GMRASLST,GMRALST,GMRAORD,GMRADUZ**  **. W !!,?2,"The following Active Orders contain "\_$P(GMRAPA(0),"^",2)\_":"**  **. S GMRADA=""**  **. F S GMRADA=$O(GMRAMCHK(GMRADA)) Q:GMRADA="" I $D(GMRAMCHK(GMRADA)) D**  **. . W !!,$P(GMRAMCHK(GMRADA),"^",3)\_" (Order #"\_$P(GMRAMCHK(GMRADA),"^")\_") An alert will be sent to:"**  **. . D GETPROV^ORWDAL32(.GMRAPROV,$P(GMRAMCHK(GMRADA),"^")) ; Need ICR**  **. . N PRVDA S PRVDA=""**  **. . F S PRVDA=$O(GMRAPROV(PRVDA)) Q:PRVDA="" I $D(GMRAPROV(PRVDA)) D**  **. . . S GMRASLST($P(GMRAMCHK(GMRADA),"^"),$P(GMRAPROV(PRVDA),";"))=GMRAPROV(PRVDA)**  **. . . W !,?3,$P(GMRAPROV(PRVDA),"^",2)**  **. . K GMRAPROV**  **. W !!**  **. S GMRAYN=0 D ASK^GMRAUTL("Do you wish to send the alert(s) to additional stakeholders? ",.GMRAOUT,.GMRAYN) Q:GMRAOUT**  **. I GMRAYN D ADDSTHR(.GMRALST)**  **. I $D(GMRASLST) S GMRAORD="" F S GMRAORD=$O(GMRASLST(GMRAORD)) Q:GMRAORD="" I $D(GMRASLST(GMRAORD)) D**  **. . K GMRAPROV**  **. . S GMRADUZ="" F S GMRADUZ=$O(GMRASLST(GMRAORD,GMRADUZ)) Q:GMRADUZ="" I $D(GMRASLST(GMRAORD,GMRADUZ)) D**  **. . . S GMRAPROV(GMRADUZ)=GMRASLST(GMRAORD,GMRADUZ)**  **. . I $D(GMRALST) M GMRAPROV=GMRALST**  **. . D SENDALRT^ORWDAL32(.GMRASEND,GMRAORD,.GMRAPROV) ; Need ICR**  **. K GMRASEND,GMRASLST**  **K GMRAMCHK**  **Q**  **ADDSTHR(GMRALST) ; NSR 20070203**  **K GMRALST**  **N I,GMRAYN S GMRAYN=0**  **F I=1:1 D Q:'GMRAYN**  **. S DIC="^VA(200,",DIC(0)="AEBQ"**  **. S DIC("A")="Enter Stakeholder's Name: " D ^DIC**  **. I +Y S GMRALST($P(Y,"^"))=$P(Y,"^")\_";VA(200,"\_"^"\_$P(Y,"^",2) K DIC,Y**  **. S GMRAYN=0 D ASK^GMRAUTL("Add another stakeholder? ",.GMRAOUT,.GMRAYN) Q:GMRAOUT**  **Q** |

###### RPC ORWDAL32 CHKMEDS

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | ‘ORWDAL32 CHKMEDS' | | |
| **TAG^RTN** | CHKMEDS^ORWDAL32 | | |
| **Input Parameters** | ORDFN (Patient Identifier), GMRAGNT (List string for added allergy) | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20070203 - Check newly entered allergy against the previously ordered medications. | | |

###### RPC ORWDAL32 GETPROV

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWDAL32 GETPROV' | | |
| **TAG^RTN** | GETPROV^ORWDAL32 | | |
| **Input Parameters** | ORDLST – List of orders on which to get provider information | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20070203 - Return the list of providers associated with the orders being checked. | | |

###### RPC ORWDAL32 SENDALRT

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWDAL32 SENDALRT' | | |
| **TAG^RTN** | SENDALRT^ORWDAL32 | | |
| **Input Parameters** | StartDate, EndDate | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20070203 - Send alert to the specified providers. | | |

##### NSR #20080704 - Enhanced Allergy Checks COTS

###### PSODRG

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

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

| 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 |
| --- |
| SET ;  N STAT S PSODRUG("IEN")=+PSOY,PSODRUG("VA CLASS")=$P(PSOY(0),"^",2)  S PSODRUG("NAME")=$P(PSOY(0),"^")  S:+$G(^PSDRUG(+PSOY,2)) PSODRUG("OI")=+$G(^(2)),PSODRUG("OIN")=$P(^PS(50.7,+$G(^(2)),0),"^")  S PSODRUG("NDF")=$S($G(^PSDRUG(+PSOY,"ND"))]"":+^("ND")\_"A"\_$P(^("ND"),"^",3),1:0)  S PSODRUG("MAXDOSE")=$P(PSOY(0),"^",4),PSODRUG("DEA")=$P(PSOY(0),"^",3)  S PSODRUG("CLN")=$S($D(^PSDRUG(+PSOY,"ND")):+$P(^("ND"),"^",6),1:0)  S PSODRUG("SIG")=$P(PSOY(0),"^",5)  I $G(PSODRUG("NDC"))="" S PSODRUG("NDC")=$$GETNDC^PSSNDCUT(+PSOY,$G(PSOSITE))  S PSODRUG("DAW")=+$$GET1^DIQ(50,+PSOY,81)  S PSODRUG("STKLVL")=$G(^PSDRUG(+PSOY,660.1))  G:$G(^PSDRUG(+PSOY,660))']"" SETX  S PSOX1=$G(^PSDRUG(+PSOY,660))  S PSODRUG("COST")=$P($G(PSOX1),"^",6)  S PSODRUG("UNIT")=$P($G(PSOX1),"^",8)  S PSODRUG("EXPIRATION DATE")=$P($G(PSOX1),"^",9)  SETX K PSOX1,PSOY  Q |

| Modified Logic (Changes are in bold) |
| --- |
| SET ; ADD ALLERGY STUFF HERE  N STAT S PSODRUG("IEN")=+PSOY,PSODRUG("VA CLASS")=$P(PSOY(0),"^",2)  S PSODRUG("NAME")=$P(PSOY(0),"^")  S:+$G(^PSDRUG(+PSOY,2)) PSODRUG("OI")=+$G(^(2)),PSODRUG("OIN")=$P(^PS(50.7,+$G(^(2)),0),"^")  S PSODRUG("NDF")=$S($G(^PSDRUG(+PSOY,"ND"))]"":+^("ND")\_"A"\_$P(^("ND"),"^",3),1:0)  S PSODRUG("MAXDOSE")=$P(PSOY(0),"^",4),PSODRUG("DEA")=$P(PSOY(0),"^",3)  S PSODRUG("CLN")=$S($D(^PSDRUG(+PSOY,"ND")):+$P(^("ND"),"^",6),1:0)  S PSODRUG("SIG")=$P(PSOY(0),"^",5)  I $G(PSODRUG("NDC"))="" S PSODRUG("NDC")=$$GETNDC^PSSNDCUT(+PSOY,$G(PSOSITE))  S PSODRUG("DAW")=+$$GET1^DIQ(50,+PSOY,81)  S PSODRUG("STKLVL")=$G(^PSDRUG(+PSOY,660.1))  G:$G(^PSDRUG(+PSOY,660))']"" SETX  S PSOX1=$G(^PSDRUG(+PSOY,660))  S PSODRUG("COST")=$P($G(PSOX1),"^",6)  S PSODRUG("UNIT")=$P($G(PSOX1),"^",8)  S PSODRUG("EXPIRATION DATE")=$P($G(PSOX1),"^",9)  **S PSODRUG(“ALLERGY CHECK”) =$P($G(PSOX1),"^",TBA)**  SETX K PSOX1,PSOY  Q |

###### PSSHREQ

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | XMLHDR^MXMLUTL  ATRIBUTE^PSSHRCOM  SITE^VASITE |

| 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 |
| --- |
| Adding the new tag ALLERGY to routine. |

| Modified Logic (Changes are in bold) |
| --- |
| **ALLERGY(PSSBASE) ;**  **; TESTING NOT READY FOR RELEASE**  **;Q**  **NEW GMRA,CNT,X,FN,NIEN,VUID,PSSXML,IDXML,ALTYXML,ALLGTYP,COTIEN,GMRAL,COTSDATA,COTSCODE,IDXXML,NODE,FDBCODE**  **SET PSSXML=""**  **IF $DATA(^TMP($J,PSSBASE,"IN","ALLERGYCHECK")) DO QUIT PSSXML ;set for PEPS Server test**  **. SET FDBCODE=$G(^TMP($J,PSSBASE,"IN","ALLERGYCHECK"))**  **. QUIT:FDBCODE=""**  **. SET PSSXML="<allergenInformation><allergenType>ingredient</allergenType><identifier>"\_FDBCODE\_"</identifier></allergenInformation>"**  **. SET PSSXML="<allergyCheck>"\_PSSXML\_"</allergyCheck>"**  **;**  **;IF PSSBASE="PSOPEPS" DO ;maybe not...need to check this**  **SET GMRA="0^0^111^1"**  **DO EN2^GMRADPT**  **;**  **Q:('$D(GMRAL)) PSSXML**  **;**  **SET X="",CNT="",NODE="^TMP($JOB,PSSBASE,""OUT"",""ALLERGYSAVE"")",FDBNODE="^TMP($JOB,""FDB PROCESSED"")"**  **KILL @NODE,@FDBNODE**  **FOR SET X=$O(GMRAL(X)) QUIT:(X="") DO**  **. SET FN=+$P($P($G(GMRAL(X)),"^",9),"(",2)**  **. SET NIEN=+$P($G(GMRAL(X)),"^",9)**  **. SET VUID=$$GET1^DIQ(FN,NIEN\_",",99.99)**  **. QUIT:(VUID="")**  **. QUIT:'$DATA(^GMRD(120.89,"B",VUID)) ;NEED TO GET PERMISSION TO USE FILE I CREATED**  **. SET COTIEN=$ORDER(^GMRD(120.89,"B",VUID,""))**  **. SET ALLGTYP=$$GET1^DIQ(120.89,COTIEN,3)**  **. QUIT:ALLGTYP=""**  **. ;QUIT:ALLGTYP["GENERIC" ;TYPE WILL CAUSE THE INTERFACE TO FAIL**  **. SET ALLGTYP=$$LOW^XLFSTR(ALLGTYP)**  **. SET COTSCODE=$$GET1^DIQ(120.89,COTIEN,2)**  **. QUIT:COTSCODE=""**  **. SET COTSDATA(ALLGTYP,COTSCODE)=VUID**  **. M @NODE@(COTSCODE)=GMRAL(X) ;SAVE PATIENT ALLERGY DATA FOR DISPLAY**  **. S ^TMP($J,COTS,COTSCODE)=””**  **. ;S @FDBNODE@(X)=GMRAL(X) ;SAVE ALLERGENS THAT WILL BE PROCESSED BY FDB**  **;**  **NEW ALLGTYP,COTSCODE**  **SET ALLGTYP=""**  **FOR SET ALLGTYP=$ORDER(COTSDATA(ALLGTYP)) QUIT:(ALLGTYP="") DO**  **. SET ALTYXML="<allergenType>"\_ALLGTYP\_"</allergenType>"**  **. SET (IDXXML,COTSCODE)=""**  **. FOR SET COTSCODE=$ORDER(COTSDATA(ALLGTYP,COTSCODE)) QUIT:(COTSCODE="") DO**  **. . SET IDXXML=$G(IDXXML)\_"<allergenInformation>"\_ALTYXML\_"<identifier>"\_COTSCODE\_"</identifier>"\_"</allergenInformation>"**  **. SET PSSXML=$G(PSSXML)\_IDXXML**  **QUIT:$G(PSSXML)="" PSSXML**  **SET PSSXML="<allergyCheck>"\_PSSXML\_"</allergyCheck>"**  **QUIT PSSXML QUIT PSSXML**  **;;**  **ALLERGY2(PSSBASE)**- This testing code, once verified will be updated to standards  NEW GMRA,CNT,X,FN,NIEN,VUID,PSSXML,IDXML,ALTYXML,ALLGTYP,COTIEN,GMRAL,COTSDATA,COTSCODE,IDXXML,NODE,FDBCODE  SET PSSXML=""  IF $DATA(^TMP($J,PSSBASE,"IN","ALLERGYCHECK")) DO QUIT PSSXML ;set for PEPS Server test  . SET FDBCODE=$G(^TMP($J,PSSBASE,"IN","ALLERGYCHECK"))  . QUIT:FDBCODE=""  . SET PSSXML="<allergenInformation><allergenType>ingredient</allergenType><identifier>"\_FDBCODE\_"</identifier></allergenInformation>"  . SET PSSXML="<allergyCheck>"\_PSSXML\_"</allergyCheck>"  ;  ;IF PSSBASE="PSOPEPS" DO  SET GMRA="0^0^111^1"  DO EN2^GMRADPT  ;  Q:('$D(GMRAL)) PSSXML  ;  S GMRAOTH=+$O(^GMRD(120.83,"B","OTHER REACTION",0)) ;GET PERMISSION  D GETDATA^GMRAOR(DFN) ;GET PERMISSION  Q:'$D(^TMP("GMRAOC",$J))  SET X="",CNT="",NODE="^TMP($JOB,PSSBASE,""OUT"",""ALLERGYSAVE"")",FDBNODE="^TMP($JOB,""FDB PROCESSED"")"  KILL @NODE,@FDBNODE  S A=""  F S A=$O(^TMP("GMRAOC",$J,A)) Q:A="" D  .S X=""  .F S X=$O(^TMP("GMRAOC",$J,A,X)) Q:X="" D  ..S Y=""  ..S Y=$O(^TMP("GMRAOC",$J,A,X,Y)) Q:Y="" D  ...S IEN=$P($P(^TMP("GMRAOC",$J,A,X,Y),"^",7),";",1)  ...S FN=$P($P($P($P(^TMP("GMRAOC",$J,A,X,Y),"^",7),";",2),"(",2),",")  ...SET VUID=$$GET1^DIQ(FN,IEN\_",",99.99)  ...QUIT:(VUID="")  ...QUIT:'$DATA(^GMRD(120.89,"B",VUID))  ...SET COTIEN=$ORDER(^GMRD(120.89,"B",VUID,"")),COTSCODE=$$GET1^DIQ(120.89,COTIEN,2)  ...SET ALLGTYP=$$GET1^DIQ(120.89,COTIEN,3)  ...QUIT:ALLGTYP=""  ...SET ALLGTYP=$$LOW^XLFSTR(ALLGTYP)  ...SET COTSCODE=$$GET1^DIQ(120.89,COTIEN,2)  ...QUIT:COTSCODE=""  ...SET COTSDATA(ALLGTYP,COTSCODE)=VUID  ...M @NODE@(COTSCODE)=GMRAL(Y) ;SAVE PATIENT ALLERGY DATA FOR DISPLAY  ;  NEW ALLGTYP,COTSCODE  SET ALLGTYP=""  FOR SET ALLGTYP=$ORDER(COTSDATA(ALLGTYP)) QUIT:(ALLGTYP="") DO  . SET ALTYXML="<allergenType>"\_ALLGTYP\_"</allergenType>"  . SET (IDXXML,COTSCODE)=""  . FOR SET COTSCODE=$ORDER(COTSDATA(ALLGTYP,COTSCODE)) QUIT:(COTSCODE="") DO  . . SET IDXXML=$G(IDXXML)\_"<allergenInformation>"\_ALTYXML\_"<identifier>"\_COTSCODE\_"</identifier>"\_"</allergenInformation>"  . SET PSSXML=$G(PSSXML)\_IDXXML  QUIT:$G(PSSXML)="" PSSXML  SET PSSXML="<allergyCheck>"\_PSSXML\_"</allergyCheck>"  QUIT PSSXML |

###### PSSHREQ

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | XMLHDR^MXMLUTL  ATRIBUTE^PSSHRCOM  SITE^VASITE |

| 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 |
| --- |
| CHECKS(PSSBASE) ;  ; @DESC Builds the checks PSSXML element  ;  ; @PSSBASE Base of input global  ;  ; @RETURNS An PSSXML string representing the checks element.  ;  .  .  ; OPTIONAL. TBA Right now set to false, will be used in future  SET PSS("useCustomTables")=$$ATRIBUTE^PSSHRCOM("useCustomTables","true")  ;  SET PSSXML="<checks"  SET PSSXML=PSSXML\_" "\_$GET(PSS("prospectiveOnly"))  SET PSSXML=PSSXML\_" "\_$GET(PSS("useCustomTables"))  SET PSSXML=PSSXML\_" >"  SET PSSXML=PSSXML\_$$CHEKDOSE(PSSBASE)  SET PSSXML=PSSXML\_$$CHEKDRUG(PSSBASE)  SET PSSXML=PSSXML\_$$CHEKTHER(PSSBASE)  SET PSSXML=PSSXML\_"</checks>"  ;  ; Return the full drugCheck element  QUIT PSSXML |

| Modified Logic (Changes are in bold) |
| --- |
| CHECKS(PSSBASE) ;  ; @DESC Builds the checks PSSXML element  ;  ; @PSSBASE Base of input global  ;  ; @RETURNS An PSSXML string representing the checks element.  ;  .  .  ; OPTIONAL. TBA Right now set to false, will be used in future  SET PSS("useCustomTables")=$$ATRIBUTE^PSSHRCOM("useCustomTables","true")  ;  SET PSSXML="<checks"  SET PSSXML=PSSXML\_" "\_$GET(PSS("prospectiveOnly"))  SET PSSXML=PSSXML\_" "\_$GET(PSS("useCustomTables"))  SET PSSXML=PSSXML\_" >"  SET PSSXML=PSSXML\_$$CHEKDOSE(PSSBASE)  SET PSSXML=PSSXML\_$$CHEKDRUG(PSSBASE)  SET PSSXML=PSSXML\_$$ALLERGY^PSSHREQ(PSSBASE) ;#205  SET PSSXML=PSSXML\_$$CHEKTHER(PSSBASE)  SET PSSXML=PSSXML\_"</checks>"  ;  ; Return the full drugCheck element  QUIT PSSXML |

###### PSSHRIT

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | XMLHDR^MXMLUTL  ATRIBUTE^PSSHRCOM  SITE^VASITE |

| 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 |
| --- |
| CUSTOM() ; check custom drug-drug interaction  ; Return 1 if OK, 0 if not OK.  ;  N INFO,INTRO,ORDER,DRUG1,DRUG2,BASE,STATUS,PSSPEC,ALLRGEN ;#205  N PSSLEFT S PSSLEFT=4 ; left margin for results  S BASE=$T(+0)\_" CUSTOM"  S ORDER="Z;1;PROSPECTIVE;1"  S DRUG1="CLARITHROMYCIN 250MG TAB",DRUG2="DIAZEPAM 5MG TAB"  K ^TMP($J,BASE)  S ^TMP($J,BASE,"IN","DRUGDRUG")=""  S ^TMP($J,BASE,"IN","PROSPECTIVE","Z;1;PROSPECTIVE;1")="16373^4010075F^^CLARITHROMYCIN 250MG TAB"  S ^TMP($J,BASE,"IN","PROFILE","I;10U;PROFILE;10")="3768^40002216^^DIAZEPAM 5MG TAB"  S ^TMP($J,BASE,"IN","ALLERGYCHECK")=3608 ;#205 FDB code for a claritromycin allergy from file 120.89  D IN^PSSHRQ2(BASE)  ;  S STATUS=$G(^TMP($J,BASE,"OUT",0))  S INTRO="Performing Custom Drug-Drug Interaction Order Check for "\_DRUG1\_" and "\_DRUG2  D SCUST  S INTRO=INTRO\_$S($L(INFO):"...OK",STATUS=0:"...OK",1:"...Not OK")  I '$L(INFO) D  .D OUTPUT(INTRO)  .I STATUS'=0 W ! D OUTPUT("Custom Drug-Drug Interaction Order Check could not be performed.",PSSLEFT)  E D  . W !  . D OUTPUT(INTRO)  . W !  . S PSSPEC("CLINICAL EFFECTS: ")=""  . S INFO=$$REPLACE^XLFSTR(INFO,.PSSPEC)  .S INFO="Significant Drug Interaction: "\_INFO  .D OUTPUT(INFO,PSSLEFT)  W !  Q $S(STATUS=0:1,$L(INFO)=0:0,1:1) |

| Modified Logic (Changes are in bold) |
| --- |
| RUNTEST ; run interaction test to PEPS server  ; called from PSS CHECK PEPS SERVICES SETUP option  D KILL^XUSCLEAN  N STATUS,X,PSSFLAG,%ZIS,POP,ZTDESC,ZTQUEUED,ZTREQ,ZTRTN  S PSSFLAG=ION  ;  W !!,"This option performs several checks. You may queue this report if you wish."  W !!,"Among these checks are:"  W !,"-----------------------"  W !,"A connection check to the Vendor Database"  W !,"Drug-Drug Interaction Check"  W !,"Duplicate Therapy Order Check"  W !,"Dosing Order Check"  **W !,"Custom Drug-Drug Interaction and Allergy Check" ;#205**  CUSTOM()  .  E D  . W !  . D OUTPUT(INTRO)  . W !  . S PSSPEC("CLINICAL EFFECTS: ")=""  . S INFO=$$REPLACE^XLFSTR(INFO,.PSSPEC)  .S INFO="Significant Drug Interaction: "\_INFO  .D OUTPUT(INFO,PSSLEFT)  **.I $D(^TMP($J,BASE,"OUT","ALLERGYCHECK",1,"MESSAGE")) D ;#205**  **..W !**  **..S INFO=^TMP($J,BASE,"OUT","ALLERGYCHECK",1,"MESSAGE")**  **..S ALLRGEN=^TMP($J,BASE,"OUT","ALLERGYCHECK",1,"ALLDESC")**  **..S INFO="ALLERGY reaction to "\_ALLRGEN\_" : "\_INFO**  **..D OUTPUT(INFO,PSSLEFT) ;#205 END**  W !  Q $S(STATUS=0:1,$L(INFO)=0:0,1:1) |
|  |

###### PSSHRQ20

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSSHRQ20 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20080704 - Enhanced Allergy Checks COTS | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| SELTAG(NAME) ;  ; @DESC Returns the appropriate tag to handle the XML element  ;  ; @NAME Name of the current XML element  ;  ; @RETURNS Returns the appropriate tag to handle the XML element  ;  QUIT:NAME="drugsNotChecked" "DRGNTCK^PSSHRQ23"  ;  QUIT:NAME="drugDrugChecks" "DRUGDRUG^PSSHRQ21"  ;  QUIT:NAME="drugTherapyChecks" "THERAPY^PSSHRQ22"  ;  QUIT:NAME="drugDoseChecks" "DRGDOSE^PSSHRQ23"  ;  QUIT  ; |

| Modified Logic (Changes are in bold) |
| --- |
| SELTAG(NAME) ;  ; @DESC Returns the appropriate tag to handle the XML element  ;  ; @NAME Name of the current XML element  ;  ; @RETURNS Returns the appropriate tag to handle the XML element  ;  QUIT:NAME="drugsNotChecked" "DRGNTCK^PSSHRQ23"  ;  QUIT:NAME="drugDrugChecks" "DRUGDRUG^PSSHRQ21"  ;  QUIT:NAME="drugTherapyChecks" "THERAPY^PSSHRQ22"  ;  QUIT:NAME="drugDoseChecks" "DRGDOSE^PSSHRQ23"  ;  **QUIT:NAME="allergyChecks" "ALLGYCHK^PSSHRQ21" ;205**  ;  QUIT  ; |

###### PSSHRQ21

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSSHRQ21 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20080704 - Enhanced Allergy Checks COTS | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | PSSHRQ20 |  |

| 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 |
| --- |
| New tags |

| Modified Logic (Changes are in bold) |
| --- |
| **ALLGYCHK(DOCHAND,NODE,BASE) ; NEW TAG #205**  **; @DESC Handles putting the allergyChecks XML element into**  **; the DrugOrderChecks object**  **;**  **; @DOCHAND Handle to XML document**  **; @NODE Node associated with XML element**  **; @PSSHAND Handle to allergenCheck object**  **;**  **; @RETURNS Nothing**  **NEW PSS,ALGYHASH**  **;**  **SET PSS("child")=0**  **SET PSS("allergyCount")=0**  **;**  **FOR SET PSS("child")=$$CHILD^MXMLDOM(DOCHAND,NODE,PSS("child")) QUIT:PSS("child")=0 DO**  **. SET PSS("childName")=$$NAME^MXMLDOM(DOCHAND,PSS("child"))**  **.**  **. SET:PSS("childName")="allergenCheck" PSS("allergyCount")=PSS("allergyCount")+1**  **. DO:PSS("childName")="allergenCheck" ALGYREAD(DOCHAND,PSS("child"),PSS("allergyCount"),BASE,.ALGYHASH)**  **. QUIT**  **;**  **DO ALGYWRIT(.ALGYHASH,BASE) ;UNCOMMENTED THIS OUT AND SENDING .ALGYHAST TO ALGYREAD UP ABOVE**  **DO ALGYOUT(BASE) ;MOVED HERE TO TEST**  **QUIT**  **ALGYREAD(DOCHAND,NODE,COUNT,BASE,ALGYHASH) ; NEW TAG #205**  **; @DESC Handles parsing and storage of allergyChecks element**  **; @DOCHAND Handle to XML document**  **; @NODE Node associated with XML element**  **; @HASH Where to store info (by ref)**  **; @COUNT Count of allergy matches sections**  **;**  **; @RETURNS Nothing**  **;QUIT ;TESTING**  **NEW PSS ;,ALGYHASH COMMENT OUT ALGYHASH AND ADDED TO PARAMETER LIST ABOVE**  **;**  **SET PSS("child")=0**  **;**  **FOR SET PSS("child")=$$CHILD^MXMLDOM(DOCHAND,NODE,PSS("child")) QUIT:PSS("child")=0 DO**  **. SET PSS("childName")=$$NAME^MXMLDOM(DOCHAND,PSS("child"))**  **. ;**  **. SET:PSS("childName")="screenMessage" ALGYHASH(COUNT,"screenMessage")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))**  **. ;**  **. DO:PSS("childName")="drug" PARSEDRG^PSSHRCOM(DOCHAND,PSS("child"),.ALGYHASH,COUNT)**  **.**  **. SET:PSS("childName")="drugDescription" ALGYHASH(COUNT,"drugDescription")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))**  **.**  **. SET:PSS("childName")="allergenDescription" ALGYHASH(COUNT,"allergenDescription")=$$UPPER^PSSHRCOM($$GETTEXT^PSSHRCOM(DOCHAND,PSS("child")))**  **. ;**  **. SET:PSS("childName")="matchDescription" ALGYHASH(COUNT,"matchDescription")=$$UPPER^PSSHRCOM($$GETTEXT^PSSHRCOM(DOCHAND,PSS("child")))**  **. ;**  **. DO:PSS("childName")="screenAllergy" PARSEMSG^PSSHRCOM(DOCHAND,PSS("child"),.ALGYHASH,COUNT)**  **. ;**  **. SET:PSS("childName")="classification" ALGYHASH(COUNT,"classification")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))**  **. ;**  **QUIT**  **ALGYWRIT(HASH,BASE) ;NEW TAG #205**  **; @DESC Handles writing allergyChecks section of the XML document**  **;**  **; @HASH ByRef, Hash used to store response**  **; @BASE Base of output global**  **; @RETURNS Nothing. Stores values in output global.**  **;**  **NEW I,NODE**  **;**  **; Loop through the allergyCheck elements**  **SET I=""**  **FOR SET I=$ORDER(HASH(I)) QUIT:I="" DO**  **. ;**  **. ; Create the node to use with subscript indirection**  **. SET NODE="^TMP($JOB,BASE,""OUT"",""ALLERGYCHECK"")" ;REMOVE DRUGDRUG SUBSCRIPT**  **. ;**  **. I $$CHKHASH(.HASH,I,"allergenDescription") SET @NODE@(I,"ALLDESC")=HASH(I,"allergenDescription")**  **. I $$CHKHASH(.HASH,I,"drugDescription") SET @NODE@(I,"DRUGDESC")=HASH(I,"drugDescription")**  **. I $$CHKHASH(.HASH,I,"drugName") SET @NODE@(I,"DRUGNAME")=HASH(I,"drugName")**  **. I $$CHKHASH(.HASH,I,"matchDescription") SET @NODE@(I,"MATCHDESC")=HASH(I,"matchDescription")**  **. I $$CHKHASH(.HASH,I,"id") SET @NODE@(I,"MATCHID")=HASH(I,"id")**  **. I $$CHKHASH(.HASH,I,"screenMessage") SET @NODE@(I,"MESSAGE")=HASH(I,"screenMessage")**  **. I $$CHKHASH(.HASH,I,"vuid") SET @NODE@(I,"VUID")=HASH(I,"vuid")**  **. I $$CHKHASH(.HASH,I,"classification") SET @NODE@(I,"CLASS")=HASH(I,"classification")**  **QUIT**  **ALGYOUT(BASE) ; NEW TAG #205**  **;**  **QUIT:$DATA(^TMP($J,BASE,"IN","ALLERGYCHECK")) ;Quit if this is a PEPS server check**  **;**  **NEW LINE,NODE,COTSNUM,J,K,L,DIWF,DIWL,DIWR,SAVENODE,WRITNODE,X,COMMA,TEMPSYM,STARS,DIR,LINE,LINE2,VUID**  **SET $P(LINE,"-",80)="-",COMMA="",TEMPSYM="",$P(STARS,"\*",79)="\*",$P(LINE2,"=",79)="="**  **; Create the node to use with subscript indirection REMOVED DRUGDRUG SUBSCRIPT**  **SET NODE="^TMP($JOB,BASE,""OUT"",""ALLERGYCHECK"")",SAVENODE="^TMP($JOB,BASE,""OUT"",""ALLERGYSAVE"")"**  **SET WRITNODE="^UTILITY($J,""W"",1)"**  **;**  **IF '$D(@NODE) Q ;Quit if there is nothing to print**  **;**  **DO:(BASE'="PSOPEPS") FULL^VALM1**  **; Loop through the allergyCheck elements**  **SET J=""**  **FOR SET J=$ORDER(@NODE@(J)) QUIT:J="" DO**  **. SET COTSNUM=$G(@NODE@(J,"MATCHID")) ;**  **. SET VUID=$G(@NODE@(J,"VUID"))**  **. W $C(7),!!,"A Drug-Allergy Reaction exists for this medication and/or class! (C)",!!**  **. IF $G(@NODE@(J,"DRUGNAME"))'="" W ?6,"Prospective Drug: ",@NODE@(J,"DRUGNAME"),!**  **. IF $G(@NODE@(J,"ALLDESC"))'="" W ?7,"Causative Agent: ",@NODE@(J,"ALLDESC"),!**  **. IF $G(@NODE@(J,"id"))'="" D ;changed MATCHID to id**  **. . SET COTSNUM=$G(@NODE@(J,"id")) ;NEW 21DEC2016 changed MATCHID to id**  **. IF COTSNUM'="" DO**  **. . IF $P($P(@SAVENODE@(COTSNUM),"^",10),";")'="" WRITE ?3,"Historical/Observed: ",$P($P(@SAVENODE@(COTSNUM),"^",10),";"),! ;NEW 21DEC2016**  **. . ;IF $PIECE($G(@SAVENODE@(COTSNUM,"O",1)),";")'="" WRITE ?14,"Severity: ",$PIECE($G(@SAVENODE@(COTSNUM,"O",1)),";"),!**  **. . IF $PIECE($G(@SAVENODE@(COTSNUM,"H")),";")'="" WRITE ?14,"Severity: ",$PIECE($G(@SAVENODE@(COTSNUM,"H")),";"),!**  **. IF $G(@NODE@(J,"MATCHDESC"))'="" D**  **. . W ?1,"FDB match description: "**  **. . SET X=$G(@NODE@(J,"MATCHDESC"))**  **. . IF $G(COTSNUM)'="",$G(@NODE@(J,"MATCHDESC"))'="" SET X=X\_" (id:"\_+COTSNUM\_")" ;NEW 21DEC2016**  **. . ;SET X=X\_" (VUID:"\_VUID\_")" ;NEW 21DEC2016**  **. . KILL @WRITNODE**  **. . SET DIWL=1,DIWR=55,DIWF=""**  **. . DO ^DIWP F L=1:1:@WRITNODE WRITE ?24,@WRITNODE@(L,0),!**  **. . SET K=""**  **. . KILL @WRITNODE**  **. . FOR SET K=$O(@SAVENODE@(COTSNUM,"S",K)) QUIT:(K="") S:K'=1 COMMA=", " S TEMPSYM=$P(@SAVENODE@(COTSNUM,"S",K),";")\_COMMA\_TEMPSYM ;21DEC2016**  **. . IF TEMPSYM'="" W !,?8,"Signs/Symptoms: " S X=TEMPSYM S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!**  **. KILL @WRITNODE**  **. IF $G(@NODE@(J,"CLASS"))'="" W !,?6,"Drug Class(C): " SET X=@NODE@(J,"CLASS") S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!**  **. KILL @WRITNODE**  **. IF $G(@NODE@(J,"MESSAGE"))'="" W !,?9,"Alert message: " SET X=@NODE@(J,"MESSAGE") S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!**  **. NEW COTSNUM,TEMPSYM ;**  **. WRITE !**  **. SET DIR(0)="E",DIR("?")="Press Return to continue",DIR("A")="Press Return to continue" DO ^DIR KILL DIR**  **Q** |

###### PSSHRQ26

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSSHRQ26 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20080704 - Enhanced Allergy Checks COTS | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | PSSHRQ2O |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| ALLGYCHK(DOCHAND,NODE,BASE) ; NEW TAG #205  ; @DESC Handles putting the allergyChecks XML element into  ; the DrugOrderChecks object  ;  ; @DOCHAND Handle to XML document  ; @NODE Node associated with XML element  ; @PSSHAND Handle to allergenCheck object  ;  ; @RETURNS Nothing  NEW PSS,ALGYHASH  ;  SET PSS("child")=0  SET PSS("allergyCount")=0  ;  FOR SET PSS("child")=$$CHILD^MXMLDOM(DOCHAND,NODE,PSS("child")) QUIT:PSS("child")=0 DO  . SET PSS("childName")=$$NAME^MXMLDOM(DOCHAND,PSS("child"))  .  . SET:PSS("childName")="allergenCheck" PSS("allergyCount")=PSS("allergyCount")+1  . DO:PSS("childName")="allergenCheck" ALGYREAD(DOCHAND,PSS("child"),PSS("allergyCount"),BASE,.ALGYHASH)  . QUIT  ;  DO ALGYWRIT(.ALGYHASH,BASE) ;UNCOMMENTED THIS OUT AND SENDING .ALGYHAST TO ALGYREAD UP ABOVE  DO ALGYOUT(BASE) ;MOVED HERE TO TEST  QUIT  ALGYREAD(DOCHAND,NODE,COUNT,BASE,ALGYHASH) ; NEW TAG #205  ; @DESC Handles parsing and storage of allergyChecks element  ; @DOCHAND Handle to XML document  ; @NODE Node associated with XML element  ; @HASH Where to store info (by ref)  ; @COUNT Count of allergy matches sections  ;  ; @RETURNS Nothing  ;QUIT ;TESTING  NEW PSS ;,ALGYHASH COMMENT OUT ALGYHASH AND ADDED TO PARAMETER LIST ABOVE  ;  SET PSS("child")=0  ;  FOR SET PSS("child")=$$CHILD^MXMLDOM(DOCHAND,NODE,PSS("child")) QUIT:PSS("child")=0 DO  . SET PSS("childName")=$$NAME^MXMLDOM(DOCHAND,PSS("child"))  . ;  . SET:PSS("childName")="screenMessage" ALGYHASH(COUNT,"screenMessage")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))  . ;  . DO:PSS("childName")="drug" PARSEDRG^PSSHRCOM(DOCHAND,PSS("child"),.ALGYHASH,COUNT)  .  . SET:PSS("childName")="drugDescription" ALGYHASH(COUNT,"drugDescription")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))  .  . SET:PSS("childName")="allergenDescription" ALGYHASH(COUNT,"allergenDescription")=$$UPPER^PSSHRCOM($$GETTEXT^PSSHRCOM(DOCHAND,PSS("child")))  . ;  . SET:PSS("childName")="matchDescription" ALGYHASH(COUNT,"matchDescription")=$$UPPER^PSSHRCOM($$GETTEXT^PSSHRCOM(DOCHAND,PSS("child")))  . ;  . DO:PSS("childName")="screenAllergy" PARSEMSG^PSSHRCOM(DOCHAND,PSS("child"),.ALGYHASH,COUNT)  . ;  . SET:PSS("childName")="classification" ALGYHASH(COUNT,"classification")=$$GETTEXT^PSSHRCOM(DOCHAND,PSS("child"))  . ;  QUIT  ALGYWRIT(HASH,BASE) ;NEW TAG #205  NEW I,NODE  ;  ; Loop through the allergyCheck elements  SET I=""  FOR SET I=$ORDER(HASH(I)) QUIT:I="" DO  . ;  . ; Create the node to use with subscript indirection  . SET NODE="^TMP($JOB,BASE,""OUT"",""ALLERGYCHECK"")" ;REMOVE DRUGDRUG SUBSCRIPT  . ;  . I $$CHKHASH(.HASH,I,"allergenDescription") SET @NODE@(I,"ALLDESC")=HASH(I,"allergenDescription")  . I $$CHKHASH(.HASH,I,"drugDescription") SET @NODE@(I,"DRUGDESC")=HASH(I,"drugDescription")  . I $$CHKHASH(.HASH,I,"drugName") SET @NODE@(I,"DRUGNAME")=HASH(I,"drugName")  . I $$CHKHASH(.HASH,I,"matchDescription") SET @NODE@(I,"MATCHDESC")=HASH(I,"matchDescription")  . I $$CHKHASH(.HASH,I,"id") SET @NODE@(I,"MATCHID")=HASH(I,"id")  . I $$CHKHASH(.HASH,I,"screenMessage") SET @NODE@(I,"MESSAGE")=HASH(I,"screenMessage")  . I $$CHKHASH(.HASH,I,"vuid") SET @NODE@(I,"VUID")=HASH(I,"vuid")  . I $$CHKHASH(.HASH,I,"classification") SET @NODE@(I,"CLASS")=HASH(I,"classification")  . I '$D(@NODE@(I,"CLASS")) DO  . . SET GMRACODE=$P($G(Y(0)),U,2)  . . DO IEN^PSN50P65(,GMRACODE,"GMRADATA")  . . SET X50P65=0  . . SET X50P65=$O(^TMP($J,"GMRADATA",X50P65))  . . QUIT:X50P65=0  . . SET OUTCLASS=$G(^TMP($J,"GMRADATA",X50P65,.01))\_" "\_$G(^TMP($J,"GMRADATA",X50P65,1))  . . SET @NODE@(I,"CLASS")=OUTCLASS  QUIT  ALGYOUT(BASE) ; NEW TAG #205  QUIT:$DATA(^TMP($J,BASE,"IN","ALLERGYCHECK")) ;Quit if this is a PEPS server check  NEW LINE,NODE,COTSNUM,J,K,L,DIWF,DIWL,DIWR,SAVENODE,WRITNODE,X,COMMA,TEMPSYM,STARS,DIR,LINE,LINE2,VUID  SET $P(LINE,"-",80)="-",COMMA="",TEMPSYM="",$P(STARS,"\*",79)="\*",$P(LINE2,"=",79)="="  ; Create the node to use with subscript indirection REMOVED DRUGDRUG SUBSCRIPT  SET NODE="^TMP($JOB,BASE,""OUT"",""ALLERGYCHECK"")",SAVENODE="^TMP($JOB,BASE,""OUT"",""ALLERGYSAVE"")"  SET WRITNODE="^UTILITY($J,""W"",1)"  IF '$D(@NODE) Q ;Quit if there is nothing to print  DO:(BASE'="PSOPEPS") FULL^VALM1  ; Loop through the allergyCheck elements  SET J=""  FOR SET J=$ORDER(@NODE@(J)) QUIT:J="" DO  . SET COTSNUM=$G(@NODE@(J,"MATCHID")) ;  . SET VUID=$G(@NODE@(J,"VUID"))  . W $C(7),!!,"A Drug-Allergy Reaction exists for this medication and/or class! (C)",!!  . IF $G(@NODE@(J,"DRUGNAME"))'="" W ?6,"Prospective Drug: ",@NODE@(J,"DRUGNAME"),!  . IF $G(@NODE@(J,"ALLDESC"))'="" W ?7,"Causative Agent: ",@NODE@(J,"ALLDESC"),!  . IF $G(@NODE@(J,"id"))'="" D ;changed MATCHID to id  . . SET COTSNUM=$G(@NODE@(J,"id")) ;NEW 21DEC2016 changed MATCHID to id  . IF COTSNUM'="" DO  . . IF $P($P(@SAVENODE@(COTSNUM),"^",10),";")'="" WRITE ?3,"Historical/Observed: ",$P($P(@SAVENODE@(COTSNUM),"^",10),";"),! ;NEW 21DEC2016  . . IF $PIECE($G(@SAVENODE@(COTSNUM,"H")),";")'="" WRITE ?14,"Severity: ",$PIECE($G(@SAVENODE@(COTSNUM,"H")),";"),!  . IF $G(@NODE@(J,"MATCHDESC"))'="" D  . . ;W ?1,"FDB match description: "  . . W ?7,"Ingredients (C):"  . . SET X=$G(@NODE@(J,"MATCHDESC"))  . . IF $G(COTSNUM)'="",$G(@NODE@(J,"MATCHDESC"))'="" SET X=X\_" (id:"\_+COTSNUM\_")" ;NEW 21DEC2016  . . KILL @WRITNODE  . . SET DIWL=1,DIWR=55,DIWF=""  . . DO ^DIWP F L=1:1:@WRITNODE WRITE ?24,@WRITNODE@(L,0),!  . . SET K=""  . . KILL @WRITNODE  . . FOR SET K=$O(@SAVENODE@(COTSNUM,"S",K)) QUIT:(K="") S:K'=1 COMMA=", " S TEMPSYM=$P(@SAVENODE@(COTSNUM,"S",K),";")\_COMMA\_TEMPSYM ;21DEC2016  . . IF TEMPSYM'="" W ?8,"Signs/Symptoms: " S X=TEMPSYM S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!  . KILL @WRITNODE  . IF $G(@NODE@(J,"CLASS"))'="" W ?12,"Drug Class: " SET X=@NODE@(J,"CLASS") S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!  . KILL @WRITNODE  . IF $G(@NODE@(J,"MESSAGE"))'="" W !,?9,"Alert message: " SET X=@NODE@(J,"MESSAGE") S DIWL=1,DIWR=55,DIWF="" D ^DIWP F L=1:1:@WRITNODE W ?24,@WRITNODE@(L,0),!  . NEW COTSNUM,TEMPSYM ;  . WRITE !  . SET DIR(0)="E",DIR("?")="Press Return to continue",DIR("A")="Press Return to continue" DO ^DIR KILL DIR  Q |

###### COTS ALLERGY MAPPING (#120.89 -- ^GMRD(120.89,))

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| VUID (#.01) | New file | A unique integer assigned to reference terms VHA wide. Supplied by vendor to match VHA VUIDs. |
| ALLERGEN NAME (#1) | New file | Name of allergen supplied by vendor: "SODIUM ACETATE,ANHYDROUS" |
| COTS CODE (#2) | New file | A unique code that maps the VUID to the vendor’s file |
| ALLERGEN TYPE(#3) | New file | '1' FOR INGREDIENT;  '2' FOR ALLERGEN GROUP;  '4' FOR GENERIC;  '3' FOR GCN SEQUENCE NUMBER; |

##### NSR # 20081008 - CPRS Notification Alert Processing

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORWORB | ^XQALSURO, ^XQALSUR1, |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^XTV(8992.1) | | | | |
| **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) |
| --- |
| Created new tag (GETUSER2):  GETUSER2(ROOT,XQAUSER,FRSTDATE,LASTDATE,MAXRET,PROONLY,FLAG) ;Get PROCESSED alerts rather than pending alerts  N NCNT,KEY,RECIP,RTYP,SURRFOR,SURRNM,SURRDA  S:$G(XQAUSER)'>0 XQAUSER=DUZ  S:$G(FRSTDATE)'>0 FRSTDATE=0  S:$G(LASTDATE)'>0 LASTDATE=0  S:$G(FLAG)'>0 FLAG=0  I $$ACTVSURO^XQALSURO(XQAUSER)'>0 D RETURN^XQALSUR1(XQAUSER)  S NCNT=0 K @ROOT  N X,X1,X2,X3,X4,X20,XDEF,XCKUSER,PDATE,RECIPDA,XDA,XMDA  S PDATE=FRSTDATE I PDATE'=0 S PDATE=PDATE-.000001  F S PDATE=$O(^XTV(8992.1,"PAR",PDATE)) Q:PDATE>LASTDATE!(PDATE="") D  . S RECIPDA="" F S RECIPDA=$O(^XTV(8992.1,"PAR",PDATE,RECIPDA)) Q:RECIPDA="" D  . . S XDA="" F S XDA=$O(^XTV(8992.1,"PAR",PDATE,RECIPDA,XDA)) Q:XDA="" I $D(^XTV(8992.1,XDA)) D  . . . S XMDA="" F S XMDA=$O(^XTV(8992.1,"PAR",PDATE,RECIPDA,XDA,XMDA))  Q:XMDA="" D  . . . . S X=$G(^XTV(8992.1,XDA,0)),X1=$G(^(1)),X2=$G(^(2)),X3=$G(^(3)),  X4=$G(^(4))  . . . . S X20(0)=$G(^XTV(8992.1,XDA,20,XMDA,0))  . . . . S (SURRFOR,SURRNM,RTYP,SURRDA)=""  . . . . S RECIP=$G(^XTV(8992.1,XDA,20,XMDA,1,1,0))  . . . . I $P(RECIP,U)]"" S RTYP=$G(^XTV(8992.2,$P(RECIP,U),0))  . . . . I $P(RECIP,U,2)]"" S SURRNM=$P($G(^VA(200,$P(RECIP,U,2),0)),U)  . . . . I $P(RECIP,U,3)="Y" D  . . . . . S SURRDA=$P($G(^XTV(8992.1,XDA,20,XMDA,3,1,0)),"^")  . . . . . I SURRDA]"" S SURRFOR=$P($G(^VA(200,SURRDA,0)),U)  . . . . . I RECIPDA]"" S SURRNM=$P($G(^VA(200,RECIPDA,0)),U)  . . . . Q:XQAUSER'=RECIPDA&(XQAUSER'=SURRDA)  . . . . S NCNT=NCNT+1  . . . . I MAXRET,NCNT>MAXRET Q  . . . . S KEY=$S($P(X3,U)'="":"G ",X4>1:"L ",$P(X1,U,3,4)="^ ":"I ",  1:"R "),@ROOT@(NCNT)=KEY\_$P(X1,U)\_U\_$P(X,U),@ROOT@(NCNT,"PROCESSED")=$P(X20(0),  U,2)\_U\_$P(X20(0),U,3)\_U\_$P(X20(0),U,4)\_U\_$P(X20(0),U,5)\_U\_RTYP\_U\_SURRNM\_U\_SURRFOR  I MAXRET,NCNT>MAXRET S NCNT=MAXRET  S @ROOT=NCNT  Q |

XQALDEL

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | XQALERT, XQALERT1, XQALSUR1, ORB3, ORB3FUP1, ORB3SPEC, ORCB, ORCSAVE1, ORCSIGN, ORRCXQ, ORWORB |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^XTV(8992), ^XTV(8992.1) | | | | |
| **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) |
| --- |
| **Added Logic in DELETE+7 to manually create new “PAR” cross-reference:**  S XQXX=$O(^XTV(8992.1,"B",XQAID,0)) I XQXX>0 S XQXY=$O(^XTV(8992.1,XQXX  ,20,"B",XQAUSER,0)) I XQXY>0,XQAFOUND,'$G(XQAUSERD) S $P(^XTV(8992.1,XQXX,20,XQX  Y,0),U,4)=XQADAT**,^XTV(8992.1,"PAR",XQADAT,XQAUSER,XQXX,XQXY)="" ; P662 - PAR index**  **Added Logic in DELETEA+8 to manually create new “PAR” cross-reference:**  S XQXX=$O(^XTV(8992.1,"B",XQAID,0)) I XQXX>0 S XQXY=$O(^XTV(8992.1,XQXX  ,20,"B",XQAUSER,0)) I XQXY>0,XQAFOUND,'$G(XQAUSERD) S $P(^XTV(8992.1,XQXX,20,XQX  Y,0),U,4)=XQADAT**,^XTV(8992.1,"PAR",XQADAT,XQAUSER,XQXX,XQXY)="" ; P662 - PAR index** |

XU8P662

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | XU8P662 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DIR, BMES^XPDUTL, ENALL^DIK |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^XTMP, ^XTV(8992.1), ^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 |
| --- |
| NA |

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

ORWORB

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by RPC | XQALDATA |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Added Function PROUSER (Called by RPC ORWORB PROUSER):**  PROUSER(ORY,STRTDATE,STOPDATE,MAXRET,PROONLY) ;return current user's processed notifications for a specified date range  N FWDBY  K ^TMP("ORB",$J),^TMP("ORBG",$J)  S FWDBY="Forwarded by: "  D GETUSER2^XQALDATA("^TMP(""ORB"",$J)",DUZ,STRTDATE,STOPDATE,MAXRET,PROONLY)  D USERLIST(.ORY,STRTDATE,STOPDATE)  Q  **Added Logic in USERLIST for expanded processed alert data:**  .;if this is for processed alerts, add additional data into pieces 15 through 21  .I $D(^TMP("ORB",$J,J,"PROCESSED")) D  ..S $P(^TMP("ORBG",$J,J),U,15)=^TMP("ORB",$J,J,"PROCESSED") |

ORWTPR

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

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Added in GETNOTO:  S $P(INFO,U,4)=$$GET^XPAR("USR.`"\_USER,"ORB DAYS FOR PROCESSED ALERTS",1,"Q")  S $P(INFO,U,5)=$$GET^XPAR("USR.`"\_USER,"ORB MAX PROCESSED ALERTS",1,"Q")  Added in SAVENOTO:  I $P(INFO,U,4)]"" D EN^XPAR(USER\_";VA(200,","ORB DAYS FOR PROCESSED ALERTS",1,$P(INFO,U,4),.ERR)  I $P(INFO,U,5)]"" D EN^XPAR(USER\_";VA(200,","ORB MAX PROCESSED ALERTS",1,$P(INFO,U,5),.ERR)  **Added Function GETARCHP (Called by RPC ORWTPR GETARCHP):**  GETARCHP(INFO) ; from RPC  N LIST,L  S INFO=0  D GETLST^XPAR(.LIST,"DIV^SYS^PKG","ORB ARCHIVE PERIOD",,.ERROR)  F L=1:1:LIST S INFO=$S(INFO<$P(LIST(L),U,2):$P(LIST(L),U,2),1:INFO)  Q |

##### NSR 20081206 – Add Address of Performing Lab to Reports in CPRS Health Summary

###### GMTSLRC

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRC | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | GMTSLRCE, GMTSUP |

| 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 |
| --- |
| Chem & Hematology Report written with no Site codes or address |

| Modified Logic |
| --- |
|  |
| **Added Logic to MAIN:**  (+6) K ^TMP("LRPLS",$J)  (+14) I $D(^TMP("LRPLS",$J)) D PLS  (+15) K ^TMP("LRC",$J),^TMP("LRPLS",$J)  **Update to WRTHDR:**  W "Test Name",?48,"Result",?58,"Units",?70,"Ref Range"**,?82,"Site Code"**,!  **Update to end of WRT function:**  . W ?TAB,REF  I $P(GMX,U,9)]"" W ?82,"["\_$P(GMX,U,9)\_"]"  W !  **Added Function PLS to print lab info:**  PLS ; List performing laboratories  ;  N LINE,LRPLS,X,LREDT,LREDT2,LRFND,LRHFND K ^TMP("LREDT",$J)  W !!,"Performing Lab Sitesnactive on "\_$$FMTE^XLFDT(LREDT,"M")\_")"  . . W !,LINE  . . S LINE=$$REPEAT^XLFSTR(" ",8)\_$P(LRPLS(0),"^",2)  . . W !,LINE  ;  W !  ;  K ^TMP("LRPLS",$J),^TMP("LRPLS-ADDR",$J),^TMP("LREDT",$J)  Q |

###### GMTSLRCE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRCE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| No current logic for new functionality |

| Modified Logic |
| --- |
| **Updates in NXTST (in bold):**  (+1) N RESULT,FLAG,TEST,GMPC,GMSQN,TNM,DESCR,THER,UNIT,HI,LO,CIS,GMTSLRES**,PSITE,LREDT**  (+18) S UNIT=$P(GMTSLRES,U,5),LO=$P(GMTSLRES,U,3),HI=$P(GMTSLRES,U,4)**,PSITE=$P(GMTSLRES,U,6)**  (+23) S ^TMP("LRC",$J,IDT,GMSQN)=CDT\_U\_SPEC\_U\_TNM\_U\_RESULT\_U\_FLAG\_U\_UNIT\_U\_LO\_U\_HI**\_U\_$G(PSITE)**  (+24) **; Save performing lab ien in list**  (+25)  **S LREDT=9999999-IDT**  (+26) **I PSITE]"" S ^TMP("LRPLS",$J,PSITE,LREDT)=""** |

###### GMTSLRSC

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRSC | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | GMTSLRC, GMTSLRS7 |

| 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 |
| --- |
| Cumulative Selected (SLCU) does not show site code or address |

| Modified Logic |
| --- |
|  |
| **Added in MAIN:**  (+10) I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC  (+11) K ^TMP("LRPLS",$J)  **Updates in WRTHDR (in Bold):**  (+4) **W ?26,"Lab"**  F GMI=0:1:6 D CKP^GMTSUP Q:'$D(HDR(GMI))!($D(GMTSQIT)) W ?(((8\*GMI+**33**)+(7-$L(HDR(GMI))\2))),$E(HDR(GMI),1,7)  **Updates in WRT (in Bold):**  (+9) **I GMI=0 W ?25,"["\_$P(GMX,U,6)\_"]"**  (+10) W ?(8\*TAB+**33**),$P(GMX,U,4)," ",$P(GMX,U,5)  **Updated in INVRT:**  (+9) . . S ^TMP("LRSR",$J,IX,IT)=GMI\_U\_$P(^TMP("LRS",$J,IT,IX),U,1,2)\_U\_$P(^TMP("LRS",$J,IT,IX),U,4,5)**\_U\_$P(^TMP("LRS",$J,IT,IX),U,9)** |

###### GMTSLRS7

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRS7 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Cumulative Selected 1 through 4 do not show site code or address |

| Modified Logic |
| --- |
| **Added in MAIN:**  (+8) I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC  **Updates in WRTHDR:**  (+3) **W ?26,"Lab"**  (+6) . W ?(((8\*GMI+**32**)+(7-$L(HDR(GMI))\2))),$E(HDR(GMI),1,7)  **Updates in WRT:**  (+8) **I GMI=0 W ?25,"["\_$P(GMX,U,6)\_"]"**  (+9) W ?(8\*TAB+**32**),$P(GMX,U,4)," ",$P(GMX,U,5)  **Updates in INVRT:**  (+6) F GMI=0:1:6 S IT=$O(^TMP("LRS",$J,IT)) Q:IT'>0 S IX="" F S IX=$O(^TMP("LRS",$J,IT,IX)) Q:IX="" S ^TMP("LRSR",$J,IX,IT)=GMI\_U\_$P(^TMP("LRS",$J,IT,IX),U,1,2)\_U\_$P(^TMP("LRS",$J,IT,IX),U,4,5)**\_U\_$P(^TMP("LRS",$J,IT,IX),U,9)** |

###### GMTSLRCP

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRCP | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Cytopathology report does not show site code or address |

| Modified Logic |
| --- |
| **Updated in MAIN (in bold):**  (+9) **I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC**  (+10) K ^TMP("LRCY",$J)**,^TMP("LRPLS",$J)**  **Updated in TRVRS (in bold):**  (+2 thru 7) changed single line to nested ‘do’  I IX0=0 D CKP^GMTSUP Q:$D(GMTSQIT) **D Q**  **.** W ?8,"Collected:",?19,$P(^TMP("LRCY",$J,IX,IX0),U)  **.** W ?31,"Acc:",?36,$P(^TMP("LRCY",$J,IX,IX0),U,2)  **. I $P(^TMP("LRCY",$J,IX,IX0),U,3)]"" W ?50,"Site Code: "\_$P(^TMP("LRCY",$J,IX,IX0),U,3)**  **. W !**  . Q |

###### GMTSLRPE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRPE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

| Modified Logic |
| --- |
| **Updates in CYSET (in +1-7):**  N ACC,CDT,D1,D2,D3,DA,DIC,DIQ,DR,DX,ICD,OT,SR,RELEASE,SITE,SN,X,YR**,PSITE,LREDT**  S CDT=$P(^LR(LRDFN,"CY",IX,0),U),ACC=$P(^(0),U,6),RELEASE=$P(^(0),U,11)  **S PSITE=$S($D(^LR(LRDFN,"CY",IX,"RF")):$G(^("RF")),1:"")**  **S LREDT=9999999-IX**  **I PSITE]"" S ^TMP("LRPLS",$J,PSITE,LREDT)=""**  I $D(ACC) S IX0=IX0+1  S X=CDT D REGDT4^GMTSU S CDT=X K X  S ^TMP("LRCY",$J,IX,0)=CDT\_U\_ACC**\_U\_PSITE**  S ^TMP("LRM",$J,0)=CDT\_U\_ACC\_U\_SS I $D(EXPAND) S ^TMP("LRM",$J,0)=^TMP(  "LRM",$J,0)\_U\_RDT\_U\_LOC  S $P(^TMP("LRM",$J,0),U,6)=CS\_U\_MICCOM**\_U\_PSITE** |

###### GMTSLRM

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRM | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Micro report does not show site code or address |

| Modified Logic |
| --- |
|  |
| **Update in MAIN:**  (+7) I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC  (+8) K ^TMP("LRPLS",$J)  **Updated in RDNODE:**  (+2) – expanded to multi-line ‘do’ and added bold code:  N GMABX,COM S GMABX=0 I GMZ=0 D Q  . D CKP^GMTSUP Q:$D(GMTSQIT) D  . . W ?10,"Collected:",?21,$P(^TMP("LRM",$J,GMZ),U),?43,"Acc:",?48,$P(^TMP("LRM",$J,GMZ),U,2)  **. . I $P(^TMP("LRM",$J,GMZ),U,8)]"" W ?58,"Site Code: ["\_$P(^TMP("LRM",$J,GMZ),U,8)\_"]"**  . . W ! |

###### GMTSLRME

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRME | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

| Modified Logic |
| --- |
|  |
| **Updates in XTRCT (in Bold):**  XTRCT ; Extract  N ACC,CDT,SS,CS,X,DIC,DIQ,DA,DR,MICRO,LOC,RDT,MICCOM**,PSITE,LREDT** K ^TMP  ("LRM",$J)  S X=$P(^LR(LRDFN,"MI",IX,0),U),RDT=$P(^(0),U,3),ACC=$P(^(0),U,6),LOC=$P  (^(0),U,8) D REGDTM4^GMTSU S CDT=X K X  D LABTEST($P(^LR(LRDFN,"MI",IX,0),U),ACC)  ; Get External format of site/specimen  ; collection sample, and comment  S DIC=63,DIQ="MICRO",DIQ(0)="E",DA=LRDFN,DA(63.05)=IX,DR=5,DR(63.05)=".  05;.055;.99"  D EN^DIQ1  S SS=MICRO(63.05,IX,.05,"E")  S CS=MICRO(63.05,IX,.055,"E"),MICCOM=MICRO(63.05,IX,.99,"E")  **S PSITE=$S($D(^LR(LRDFN,"MI",IX,"RF")):$G(^("RF")),1:"")**  **S LREDT=9999999-IX**  **I PSITE]"" S ^TMP("LRPLS",$J,PSITE,LREDT)=""** |

###### GMTSLRA

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRA | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Surgical Pathology report does not provide site code and address |

| Modified Logic |
| --- |
| **Updated in MAIN:**  (+11) I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC  (+12) K ^TMP("LRA",$J),^TMP("LRPLS",$J)  **Added in WRT:**  (+4) (removed ! character from end of this line)  (+5) . I $P(X,U,3)]"" W ?50,"Site: "\_$P(X,U,3)  (+6) . W ! |

###### GMTSLRAE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRAE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Changes in APSET (in bold):**  APSET ; Sets ^TMP("LRA",$J  N ACC,CDT,DA,DIC,DIQ,DR,GMW,SN,X,YR,SPP**,SPS,LREDT**  S CDT=$P(^LR(LRDFN,"SP",IX,0),U),SPP=$P(^LR(LRDFN,"SP",IX,0),U,7),SPP=$$GET1^DIQ(200,(+SPP\_","),.01),ACC=$P(^LR(LRDFN,"SP",IX,0),U,6)  **S SPS=$S($D(^LR(LRDFN,"SP",IX,"RF")):$G(^("RF")),1:"")**  I $S(+$P(^LR(LRDFN,"SP",IX,0),U)'>0:1,+$P(^(0),U,11)'>0:1,1:0) Q  I $D(ACC) S IX0=IX0+1  S X=CDT D REGDTM4^GMTSU S CDT=X K X  S ^TMP("LRA",$J,IX,0)=CDT\_U\_ACC**\_U\_SPS**  S:$L($G(SPP)) ^TMP("LRA",$J,IX,"SPP")=$G(SPP)  I $D(^LR(LRDFN,"SP",IX,.1)) S ^TMP("LRA",$J,IX,.1)="Site/Specimen"  S SN=0 F S SN=$O(^LR(LRDFN,"SP",IX,.1,SN)) Q:SN'>0 S ^TMP("LRA",$J,IX,.1,SN)=$P(^LR(LRDFN,"SP",IX,.1,SN,0),U)  I $D(^LR(LRDFN,"SP",IX,.2,0)),($P(^(0),U,3)]"") D CLHX  I $D(^LR(LRDFN,"SP",IX,1,0)),($P(^(0),U,3)]"") D GROSS  I $D(^LR(LRDFN,"SP",IX,1.1,0)),($P(^(0),U,3)]"") D MIC  I $D(^LR(LRDFN,"SP",IX,1.2,0)),($P(^(0),U,3)]"") D SUPPR  I $D(^LR(LRDFN,"SP",IX,1.3,0)),($P(^(0),U,3)]"") D FROZ  I $D(^LR(LRDFN,"SP",IX,1.4,0)),($P(^(0),U,3)]"") D SPDX  **S LREDT=9999999-IX**  **I SPS]"" S ^TMP("LRPLS",$J,SPS,LREDT)=""**  Q |

###### GMTSLRS

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRS | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Lab Tests Selected report does not provide site code and address data |

| Modified Logic |
| --- |
| **Updates in MAIN:**  (+16) **I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC**  (+17) **K ^TMP("LRS",$J),^TMP("LRPLS",$J)**  **Updated in WRTHDR:**  (+2) W ?29,"Test Name",?48,"Result",?58,"Units",?70,"Ref Range"**,?82,"Site Code"**,!  **Updates in WRT:**  (+15) . W ?58,$P(GMX,U,6),?TAB,REF **🡨removed !**  (+16) **. I $P(GMX,U,9)]"" W ?84,"["\_$P(GMX,U,9)\_"]"**  (+17) **. W !** |

###### GMTSLRSE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRSE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in CHSET:**  (+1) N RESULT,FLAG,DRWDT,SITE,SPEC,TNM,DESCR,THER,UNIT,HI,LO,GMIDT,GMTSLRES**,PSITE,LREDT**  (+11) S UNIT=$P(GMTSLRES,U,5),LO=$P(GMTSLRES,U,3),HI=$P(GMTSLRES,U,4)**,PSITE=$P(GMTSLRES,U,6)**  (+15) S ^TMP("LRS",$J,GMTSI,GMIDT)=DRWDT\_U\_$E(SPEC,1,10)\_U\_TNM\_U\_RESULT\_U\_FLAG\_U\_UNIT\_U\_LO\_U\_HI**\_U\_PSITE**  (+19) **S LREDT=9999999-IDRWDT**  (+20) **I PSITE]"" S ^TMP("LRPLS",$J,PSITE,LREDT)=""** |

###### XUMF04H

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | XUMF04H | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in ZIN+61:64:**  ;  ; -- check for changes to physical address  D PADDCK  ;    **New tag:**  PADDCK ; -- check for changes to physical address  ;  N XSTREET,XSTREET2,XCITY,XZIP,XSTATE  N XHPADD,XIENS S XHPADD=0  ;  ; -- retrieve current physical address fields  S XSTREET=$$GET1^DIQ(4,IENS,1.01)  S XSTREET2=$$GET1^DIQ(4,IENS,1.02)  S XCITY=$$GET1^DIQ(4,IENS,1.03)  S XZIP=$$GET1^DIQ(4,IENS,1.04)  S XSTATE=$$GET1^DIQ(4,IENS,.02)  ;  ; -- compare against fields in master file update  I STREET'=XSTREET S XHPADD=1  I STREET2'=XSTREET2 S XHPADD=1  I CITY'=XCITY S XHPADD=1  I ZIP'=XZIP S XHPADD=1  I STATE'=XSTATE S XHPADD=1  ;  ; -- if differences, create historical address array  I XHPADD D  . K XUADD,XUEFFDT  . S XUEFFDT(1)=DT  . S XIENS="+1,"\_IENS  . S XUADD(4.999,XIENS,.01)=XUEFFDT(1)  . S XUADD(4.999,XIENS,1)=XSTREET  . S XUADD(4.999,XIENS,1.1)=XSTREET2  . S XUADD(4.999,XIENS,1.2)=XCITY  . S XUADD(4.999,XIENS,1.3)=XSTATE  . S XUADD(4.999,XIENS,1.4)=XZIP  . D UPDATE^DIE("E","XUADD","XUEFFDT")  . K XUADD,XUEFFDT  K XSTREET,XSTREET2,XCITY,XZIP,XSTATE,XHPADD,XIENS  Q  ; |

###### XUAF4

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | XUAF4 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **New tag/API:**  HPADD(IEN,XUEDT) ; -- historical physical address based on effective date  ;  Q:$G(XUEDT)=""  N XUADD1,XUADD2,XUCITY,XUSTATE,XUZIP,XULINE  ;  ; Find valid historical address fields based on effective date  ;  Q:'$D(^DIC(4,+IEN,999))  N XUHFND,XUHDA,IENS,XUHRRY  S XUHDA="",XUHFND=0,XULINE=""  F S XUHDA=$O(^DIC(4,+IEN,999,"B",XUHDA)) Q:XUHDA=""!XUHFND I $D(^DIC(  4,+IEN,999,XUHDA,1)) D  . I XUEDT<XUHDA D  . . S IENS=XUHDA\_","\_+IEN\_","  . . D GETS^DIQ(4.999,IENS,"1:1.4","","XUHRRY")  . . S XUHFND=1,XUADD1=XUHRRY(4.999,IENS,1)  . . S XUADD2=XUHRRY(4.999,IENS,1.1),XUCITY=XUHRRY(4.999,IENS,1.2)  . . S XUSTATE=XUHRRY(4.999,IENS,1.3),XUZIP=XUHRRY(4.999,IENS,1.4)  . . S XULINE=XUADD1\_U\_XUADD2\_U\_XUCITY\_U\_XUSTATE\_U\_XUZIP  . . K XUHRRY  ;  Q XULINE  ; |

###### ORWLR1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWLR1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **REPORT+7:11**  ;D HEAD^ORWRPP1(ORDFN,PAGE,"PATIENT BLOOD BANK REPORT",$G(STATION))  ;D HURL^ORWRPP1(.ORY,ORDFN,"PATIENT BLOOD BANK REPORT",,,1)  ;modified lab header to include facility address  D LRHEAD^ORWRPP1(ORDFN,PAGE,"PATIENT BLOOD BANK REPORT",$G(STATION))  D LRHURL^ORWRPP1(.ORY,ORDFN,"PATIENT BLOOD BANK REPORT",,,1) |

###### ORWRPL

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWRPL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **DEQUE+12:16**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,ORID,$P(ORX,"^"))  . ;D HURL^ORWRPP1(.ORY,ORDFN,ORID,1,$P(ORX,"^"))  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,ORID,$P(ORX,"^"))  . D LRHURL^ORWRPP1(.ORY,ORDFN,ORID,1,$P(ORX,"^"))  **LOOP+8:12**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,TEXT)  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,TEXT)  **LOOP+20:24**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  **LOOP+32:36**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  **LOOP+42:46**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,"PATIENT ANATOMIC PATHOLOGY REPORT",$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,"PATIENT ANATOMIC PATHOLOGY REPORT")  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,"PATIENT ANATOMIC PATHOLOGY REPORT",$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,"PATIENT ANATOMIC PATHOLOGY REPORT")  **LOOP+52:56**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,"PATIENT BLOOD BANK REPORT",$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,"PATIENT BLOOD BANK REPORT")  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,"PATIENT BLOOD BANK REPORT",$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,"PATIENT BLOOD BANK REPORT")  **LOOP+65:69**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,TEXT,1)  **LOOP+76:80**  . ;D HEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . ;D HURL^ORWRPP1(.ORY,ORDFN,TEXT)  . ;modified lab header to include facility address  . D LRHEAD^ORWRPP1(ORDFN,PAGE,TEXT,$G(STATION))  . D LRHURL^ORWRPP1(.ORY,ORDFN,TEXT) |

###### ORWRPP

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWRPP | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Added the following new tag:**  LRSITE(ORSTA) ;Print Station info  N X,ORADD  I $G(ORSTA) S ORSTA=$$IEN^XUAF4(ORSTA)  S:'$L($G(ORSTA)) ORSTA=$G(DUZ(2))  S X="Report from: "\_$$GET1^DIQ(4,+ORSTA,.01,"E")\_" Station #"\_$$GET1^DIQ(4,+ORSTA,99,"E")  W !?(IOM/2-($L(X)/2)),X  S ORADD=$$PADD^XUAF4(+ORSTA)  S ORADD=$P(ORADD,"^")\_", "\_$P(ORADD,"^",2)\_", "\_$P(ORADD,"^",3)\_" "\_$P(ORADD,"^",4)  S ORADD=$E(ORADD,1,76)  W !?(IOM/2-($L(ORADD)/2)),ORADD  Q |

###### ORWRPP1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWRPP1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **LSB+6:10:**  ;D HEAD(ORDFN,PAGE,TEXT)  ;D HURL(.ORY,ORDFN,TEXT)  ;modified lab header to include facility address  D LRHEAD(ORDFN,PAGE,TEXT)  D LRHURL(.ORY,ORDFN,TEXT)  **Added the following new tags:**  LRHEAD(ORDFN,PAGE,TITLE,STATION) ; modified patient header to add facility address for lab rpts  Q:'$G(ORDFN)  N %,%H,%I,DISYS,ORAGE,ORDOB,ORHLINE,ORL,ORNP,ORPNM,ORPV,ORSEX,ORSSN,ORTS,ORWARD,VA,X,ORI  S:'$L($G(TITLE)) TITLE="PATIENT REPORT"  D PAT^ORPR03(ORDFN)  D LRSITE^ORWRPP($G(STATION))  W !,TITLE,?(IOM-$L("Page "\_PAGE)),"Page "\_PAGE  S X=ORDOB\_" ("\_ORAGE\_")"  W !,ORPNM\_" "\_ORSSN,?39,$G(ORL(0))\_$S($L($G(ORL(1))):"/"\_ORL(1),1:""),?(79-$L(X)),X  S $P(ORHLINE,"=",IOM+1)=""  W !,ORHLINE  S X="Printed: "\_$$DATE^ORU($$NOW^XLFDT,"MM/DD/CCYY HR:MIN")  W !?27,"\*\*\* WORK COPY ONLY \*\*\*",?(IOM-($L(X))-1),X  Q  LRHURL(Y,ORDFN,TITLE,FORMAT,STATION,READ) ; modified patient header to add facility address for lab rpts  ;FORMAT tells me which node to go after  N L,NOHURL,A,OUT  S OUT=0,L="",NOHURL=0  F S L=$O(@Y@(L)) Q:L="" Q:OUT D  . I $Y+4>IOSL D  .. S PAGE=PAGE+1  .. W !?27,"\*\*\* WORK COPY ONLY \*\*\* (continued...)"  .. I $G(READ),$G(IOT)'["HFS" R !,"^ TO STOP: ",A:DTIME I A["^" S OUT=1 Q  .. W @IOF  .. I $G(IOT)["HFS" S $Y=0  .. D LRHEAD(ORDFN,PAGE,$G(TITLE),$G(STATION))  .. W !,"(...continued)"  . I $G(FORMAT) D Q  .. Q:'$D(@Y@(L))  .. I NOHURL,$P(@Y@(L),"^")'="[REPORT TEXT]" Q  .. I NOHURL,$P(@Y@(L),"^")="[REPORT TEXT]" S NOHURL=0 Q  .. I $P(@Y@(L),"^")="[HIDDEN TEXT]" S NOHURL=1 Q  .. I @Y@(L)["\*\*PAGE BREAK\*\*" Q  .. W !,@Y@(L)  . Q:'$D(@Y@(L,0))  . I NOHURL,$P(@Y@(L,0),"^")'="[REPORT TEXT]" Q  . I NOHURL,$P(@Y@(L,0),"^")="[REPORT TEXT]" S NOHURL=0 Q  . I $P(@Y@(L,0),"^")="[HIDDEN TEXT]" S NOHURL=1 Q  . I @Y@(L,0)["\*\*PAGE BREAK\*\*" Q  . W !,@Y@(L,0)  W !?27,"\*\*\* WORK COPY ONLY \*\*\*"  Q |

###### LR7OB63

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OB63 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **PLS+11:14**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRX)  . S LREDT=9999999-IVDT  . S LINE=$$PLSADDR^LR7OSUM2(LRX,LREDT)  **PLS+31:34**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRPLS)  . S LREDT=9999999-IVDT  . S LINE=$$PLSADDR^LR7OSUM2(LRPLS,LREDT) |

###### LR7OGG

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OGG | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **PLS+11:14**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRX)  . S LREDT=9999999-IDT  . S LINE=$$PLSADDR^LR7OSUM2(LRX,LREDT)  **PLS+31:34**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRPLS)  . S LREDT=9999999-IDT  . S LINE=$$PLSADDR^LR7OSUM2(LRPLS,LREDT) |

###### LR7OGMG

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OGMG | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **PLS+14:17**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRPLS)  . S LREDT=9999999-IDT  . S LINE=$$PLSADDR^LR7OSUM2(LRPLS,LREDT) |

###### LR7OR2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OR2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **PLS+11:14**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRX)  . S LREDT=9999999-IVDT  . S LINE=$$PLSADDR^LR7OSUM2(LRX,LREDT)  **PLS+31:34**  . ;modified to pass effective date - NSR 20081206  . ;S LINE=$$PLSADDR^LR7OSUM2(LRPLS)  . S LREDT=9999999-IVDT  . S LINE=$$PLSADDR^LR7OSUM2(LRPLS,LREDT) |

###### LR7OSAP

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OSAP | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **W+19:21**  . ;modified to pass effective date - NSR 20081206  . ;I LRX D RL^LR7OSMZU(LRX)  . I LRX D RL^LR7OSMZU(LRX,LRI) |

###### LR7OSUM2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LR7OSUM2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **PLS+23:26**  . . . ;modified to pass effective date - NSR 20081206  . . . ;S LINE=$$PLSADDR(LRPLS)  . . . S LREDT=9999999-LRIDT  . . . S LINE=$$PLSADDR(LRPLS,LREDT)  **PLSADDR**  PLSADDR(LRPLS,LREDT) ; Performing lab name/address/CLIA  ; Call with LRPLS = ien of entry in file #4  ; (optional) LREDT = date of service for historical address - Added for NSR 20081206  **PLSADDR+13:16**  . ;next section added to find historical address if applicable - NSR 20081206  . I $G(LREDT)'="" D  . . S LRX=$$HPADD^XUAF4(LRPLS,LREDT) Q:LRX=""  . . S LINE=LINE\_"^"\_$P(LRX,U)\_" "\_$S($P(LRX,U,2)'="":$P(LRX,U,2)\_" ",1:"")\_$P(LRX,U,3)\_$S($P(LRX,U,4)'="":", ",1:"")\_$P(LRX,U,4)\_" "\_$P(LRX,U,5) |

###### LRRPL

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | LRRPL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **RETSEC+10**  S LRRVDT=$P(LRREF,",",3) ; LR\*5.2\*472  **RETSEC+33:41**  ; next section added to print historical site address, instead of current address, if applicable - NSR 20081206  S LRX=""  I LRRVDT'="" D  . S LREDT=9999999-LRRVDT  . S LRX=$$HPADD^XUAF4(LR4,LREDT) Q:LRX=""  . S LRREC=LRREC+1  . S LRREC(LRREC)=$P(LRX,U)\_" "\_$S($P(LRX,U,2)'="":$P(LRX,U,2)\_" ",1:"")\_$P(LRX,U,3)\_$S($P(LRX,U,4)'="":", ",1:"")\_$P(LRX,U,4)\_" "\_$P(LRX,U,5)  Q:LRX'=""  ; |

###### GMTSLRMB

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLRMB | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **New lines of code – MAIN+8:+9:**  I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC  K ^TMP("LRPLS",$J)  **New lines of code – RDNODE+4:+5:**  . I $P(^TMP("LRM",$J,GMTSJ),U,8)]"" W ?58,"Site Code: ["\_$P(^TMP("LRM",  $J,GMTSJ),U,8)\_"]"  . W ! |

###### GMTSLROE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLROE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in SET+2 (Updates in bold):**  N RL,TEST**,PSITE**  **Updates in SET+11:SET+13:**  I $D(IDT),$D(SN),$D(TN) S ^TMP("LRO",$J,IDT,SN\_TN)=CDT\_U\_TEST\_U\_SPEC\_U\_  URG\_U\_OS\_U\_MD\_U\_ODT\_U\_ACC\_U\_RDT\_U\_COLL\_U\_CD\_U\_$P(TST,U,7)**\_$G(PSITE)**,GMI=GMI+1  **; Save performing lab ien and date of interest in list**  **I $G(PSITE)]"" S ^TMP("LRPLS",$J,PSITE,IDT)=""**  **Updates in COLLECT+1:**  **S PSITE=$P(^LRO(69,CD,1,SN,1),"^",8)** |

###### GMTSLROB

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLROB | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in BOLD:**  GMTSLROB ; SLC/JER - Brief Lab Order ;09/29/16 13:25  ;;2.7;Health Summary;\*\*28,58**,115**\*\*;Oct 20, 1995  MAIN ; Driver  N GMW,GMX,ICD,MAX,OC,SN  I $D(GMTSNDM),(GMTSNDM>0) S MAX=GMTSNDM  E S MAX=999  D ^GMTSLROE  I '$D(^TMP("LRO",$J)) Q  **D WRTHDR**  S (ICD,OC)=0 F S ICD=$O(^TMP("LRO",$J,ICD)) Q:'ICD!(OC'<MAX) S SN=0 F  S SN=$O(^TMP("LRO",$J,ICD,SN)) Q:'SN!(OC'<MAX) D GET  D WRTFTR  K ^TMP("LRO",$J)  Q  GET ; Get Data  S GMX=^TMP("LRO",$J,ICD,SN),OC=OC+1 I ICD>GMTS1,(ICD'>GMTS2) D WRT  Q  WRTHDR ; Prints Header  D CKP^GMTSUP Q:$D(GMTSQIT) W "Collection DT",?18,"Test Name",?39,"Specimen",?51,"Urgency",?60,"Status",**?72,"Site",**!  W:'$D(GMTSOBJ) !  Q  WRT ; Writes Component  D CKP^GMTSUP Q:$D(GMTSQIT) D:GMTSNPG WRTHDR W $P(GMX,U),?18,$P($P(GMX,  U,2),";",2),?39,$E($P($P(GMX,U,3),";",2),1,10),?51,$P(GMX,U,4),?60,$P(GMX,U,5),**?72,$P(GMX,U,13),**!  Q  **WRTFTR ; Prints Performing Lab Footer**  **I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC**  **K ^TMP("LRPLS",$J)**  **Q** |

###### GMTSLROS

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLROS | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in BOLD:**  GMTSLROS ; SLC/JER,KER - Lab Order Status Summary ;10/03/16 09:36  ;;2.7;Health Summary;\*\*28,47**,115**\*\*;Oct 20, 1995  ;  MAIN ; Lab Order Status  N GMW,GMX,ICD,MAX,OC,SN  S MAX=$S(+($G(GMTSNDM))>0:+($G(GMTSNDM)),1:999)  D ^GMTSLROE I '$D(^TMP("LRO",$J)) Q  S (ICD,OC)=0 F S ICD=$O(^TMP("LRO",$J,ICD)) Q:'ICD!(OC'<MAX) S SN=0 F  S SN=$O(^TMP("LRO",$J,ICD,SN)) Q:'SN!(OC'<MAX) D GET  **D WRTFTR**  K ^TMP("LRO",$J)  Q  GET ; Get Data from ^TMP("LRO",$J  S GMX=^TMP("LRO",$J,ICD,SN),OC=OC+1  I ICD>GMTS1,(ICD'>GMTS2) D CKP^GMTSUP Q:$D(GMTSQIT) W:OC>1&'(GMTSNPG)  ! D WRT  Q  WRT ; Write Data  N GMI,TSET,TEST S TSET="",$P(GMX,U,3)=$E($P(GMX,U,3),1,10)  F GMI=1:1:3 S $P(TEST,"-",GMI)=$S(GMI=3:$P(GMX,U,GMI+1),1:$P($P(GMX,U,G  MI+1),";",2))  F Q:$L(TEST)<23 S TSET=$P(TEST,"-",$L(TEST,"-"))\_" "\_TSET,TEST=$P(TES  T,"-",1,$L(TEST,"-")-1)  D CKP^GMTSUP Q:$D(GMTSQIT) W $P(GMX,U),?18,$E(TEST,1,20),?39,"Prov: ",  $E($P($P(GMX,U,6),";",2),1,10),?56,"Ord'd: ",$P(GMX,U,7),!  D CKP^GMTSUP Q:$D(GMTSQIT) G:GMTSNPG WRT W **"Perf. Lab: ",$P(GMX,U,13),**  ?18,$E(TSET,1,20),?39,"# ",$E($P(GMX,U,8),1,15),?56,"Avail: ",$P(GMX,U,9),!  Q  **WRTFTR ; Prints Performing Lab Footer**  **I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC**  **K ^TMP("LRPLS",$J)**  **Q** |

###### GMTSLREE

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLREE | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in APSET (updates in BOLD):**  APSET ; Sets ^TMP("LREM",$J  N ACC,CDT,DA,DIC,DIQ,DR,GMW,SN,X,YR**,PSITE,IDT**  S **(**CDT**,IDT)**=$P(^LR(LRDFN,"EM",IX,0),U),ACC=$P(^(0),U,6)  I $S(+$P(^LR(LRDFN,"EM",IX,0),U)'>0:1,+$P(^(0),U,11)'>0:1,1:0) Q  I $D(ACC) S IX0=IX0+1  S X=CDT D REGDTM4^GMTSU S CDT=X K X  S ^TMP("LREM",$J,IX,0)=CDT\_U\_ACC  **I $D(^LR(LRDFN,"EM",IX,"RF")) D**  **. S PSITE=^LR(LRDFN,"EM",IX,"RF")**  **. S ^TMP("LREM",$J,IX,"RF")=PSITE**  **. S ^TMP("LRPLS",$J,PSITE,IDT)=""** |

###### GMTSLREM

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMTSLREM | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **New line of code at MAIN+10:**  **D WRTFTR**    **Updated code at WRT+3:**  . W ?8,"Collected:",?19,$P(X,U),?31,"Acc:",?36,$P(X,U,2),**?51,"Perf. Lab**  **:",?62,$G(^TMP("LREM",$J,IX,"RF")),**!  **New tag - WRTFTR:**  **WRTFTR ; Prints Performing Lab Footer**  **I $D(^TMP("LRPLS",$J)) D PLS^GMTSLRC**  **K ^TMP("LRPLS",$J)**  **Q** |

###### ORDV02

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORDV02 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20081206 Add Address of Performing Lab to Reports in CPRS Health Summary | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates in CH+39:CH+48:**  **... S ^TMP("ORDATA",$J,ORI,TST,"WP",10)="10^"\_$P(ORX0,U,9) ;perf lab**  **... N PSITE,PCNT,PLINE S PSITE=$P(ORX0,U,9),PCNT=1 I PSITE'="" D**  **.... I $D(^TMP("LRC",$J,ORI,"C")) S PCNT=$O(^TMP("LRC",$J,ORI,"C",-1))+1**  **.... S PLINE=$$PLSADDR(PSITE,(9999999-ORI))**  **.... S ^TMP("LRC",$J,ORI,"C",PCNT)="",PCNT=PCNT+1**  **.... S ^TMP("LRC",$J,ORI,"C",PCNT)="Performing Lab Site:",PCNT=PCNT+1**  **.... S ^TMP("LRC",$J,ORI,"C",PCNT)=$$LJ^XLFSTR("["\_PSITE\_"] ",8)\_$P(PLI**  **NE,"^"),PCNT=PCNT+1**  **.... S ^TMP("LRC",$J,ORI,"C",PCNT)=$$REPEAT^XLFSTR(" ",8)\_$P(PLINE,"^",**  **2),PCNT=PCNT+1**  ... D SPMRG^ORDVU("^TMP(""LRC"","\_$J\_","\_ORI\_",""C"")","^TMP(""ORDATA""  ,"\_$J\_","\_ORI\_","\_TST\_",""WP"",**11**)",**11**) ;comments  ... I $O(^TMP("LRC",$J,ORI,"C",0)) S ^TMP("ORDATA",$J,ORI,TST,"WP",**12**)=  "12^[+]" ;flag for details    **New tag - PLSADDR:**  **PLSADDR(ORPLS,OREDT) ; Performing lab name/address/CLIA (Added for NSR 20081206)**  **; Call with ORPLS = ien of entry in file #4**  **; (optional) OREDT = date of service for historical address**  **; Returns LINE = name [CLIA# nnnn] ^ address of institution**  **;**  **N CLIA,LINE,ORX**  **S LINE="",ORX=""**  **I LINE="" D**  **. S LINE=$$NAME^XUAF4(ORPLS),CLIA=$$ID^XUAF4("CLIA",ORPLS)**  **. I CLIA'="" S LINE=LINE\_" [CLIA# "\_CLIA\_"]"**  **. ;next section added to find historical address, if applicable**  **. I $G(OREDT)'="" D**  **. . S ORX=$$HPADD^XUAF4(ORPLS,OREDT) Q:ORX=""**  **. . S LINE=LINE\_"^"\_$P(ORX,U)\_" "\_$S($P(ORX,U,2)'="":$P(ORX,U,2)\_" ",1:**  **"")\_$P(ORX,U,3)\_$S($P(ORX,U,4)'="":", ",1:"")\_$P(ORX,U,4)\_" "\_$P(ORX,U,5)**  **. I ORX="" D**  **. . S ORX=$$PADD^XUAF4(ORPLS),ORX(1)=$$WHAT^XUAF4(ORPLS,1.02)**  **. . S LINE=LINE\_"^"\_$P(ORX,U)\_" "\_$S(ORX(1)'="":ORX(1)\_" ",1:"")\_$P(ORX**  **,U,2)\_$S($P(ORX,U,3)'="":", ",1:"")\_$P(ORX,U,3)\_" "\_$P(ORX,U,4)**  **Q LINE** |

##### NSR# 20100706 (Identify Required Fields in Text Integration Utility (TIU) Note Templates)

###### ORWTIU

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DIE FILE^DIE |

| 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: ACTION  Definition: This variable lets the software know if the action is to save Template Required Fields preferences or to load (return) them. Expected values are SVPREF = SAVE, LDPREF = LOAD | | | | |
| **Input Attribute Name and Definition** | Name: INPUT  Definition: This is the template required fields preferences to be saved. When the ACTION parameter contains LDPREF, then this value must be included.    HIGHLIGHT ON^HIGHLIGHT COLOR^NAVIGATION LOCATION    HIGHLIGHT ON = 1 (ON) OR 0 (OFF)  HIGHLIGHT COLOR = A NUMBER REPRESENTING THE  COLOR FOR HIGHLIGHTS  NAVIGATION LOCATION = THIS IS THE LOCATION OF  THE NAVIGATION CONTROLS  USED TO MOVE BETWEEN  REQUIRED FIELDS.  1 = TOP, 2 = LEFT,  3 = RIGHT, 4 = BOTTOM | | | | |
| **Output Attribute Name and Definition** | Name: VAL  Definition:  IF ACTION = LDPREF THEN:    HIGHLIGHT ON^HIGHLIGHT COLOR^NAVIGATION LOCATION    IF ACTION = SVPREF THEN:    BOOLEAN - 1 = SUCCESS  0 = FAIL | | | | |
|  |  | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **REQDFLD(VAL,ACTION,INPUT) ;Load or Save Template Required Fields Preferences**  **N COLOR,DA,HILITEON,NAVLOC**  **S VAL=0**  **;I ((ACTION="LDPREF")!(ACTION="SVPREF")) S VAL=$$RQPREF^TIUPREF(ACTION,INPUT)**  **I DUZ'>0 S VAL="-1^Invalid user" Q**  **I ACTION="SVPREF" D Q**  **. N HILITEON,COLOR,NAVLOC**  **. I INPUT="" S VAL="-1^Save data not received" Q**  **. S HILITEON=$P(INPUT,U,1)**  **. I HILITEON'=0,HILITEON'=1 S HILITEON=0 ;Default to Highligh Off if bad data received**  **. S COLOR=$P(INPUT,U,2)**  **. S NAVLOC=$P(INPUT,U,3)**  **. I ((+NAVLOC<0)!(+NAVLOC>3)) S NAVLOC=0 ;Default to Navigation bar at top if bad data received**  **. S DA=+$O(^TIU(8926,"B",DUZ,""))**  **. I DA>0,$D(^TIU(8926,DA)) D Q**  **.. N DIE,DR**  **.. I COLOR="" S COLOR="@"**  **.. S DIE="^TIU(8926,"**  **.. S DR=".21////^S X=HILITEON;.22////^S X=COLOR;.23////^S X=NAVLOC"**  **.. D ^DIE**  **.. S VAL=1**  **. I ((DA=0)!('$D(^TIU(8926,DA)))) D Q**  **.. N D0,DIC,X,Y**  **.. S DIC="^TIU(8926,"**  **.. S DIC(0)=""**  **.. S DIC("DR")=".21////^S X=HILITEON;.22////^S X=COLOR;.23////^S X=NAVLOC"**  **.. S X=DUZ**  **.. D FILE^DICN**  **.. I +Y>0 S VAL=1 Q**  **.. S VAL="-1^Save Failed"**  **. S VAL="-1^Save Failed"**  **I ACTION="LDPREF" D Q**  **. N DATA,IEN**  **. S IEN=+$O(^TIU(8926,"B",DUZ,""))**  **. I IEN<1 S VAL="-1^No Data Found" Q**  **. S DATA=$G(^TIU(8926,IEN,2))**  **. S HILITEON=$P(DATA,U,1)**  **. S COLOR=$P(DATA,U,2)**  **. S NAVLOC=$P(DATA,U,3)**  **. S VAL=$S(HILITEON="":1,1:HILITEON)\_U\_COLOR\_U\_$S(NAVLOC="":0,1:NAVLOC)**  **S VAL="-1^Invalid Action parameter"**  **Q** |

##### NSR# 20071211 – Changes to Allergy Pharmacy Package

###### ORWDXC

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDXC | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20071211 (Changes to Allergy Pharmacy Package) | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by RPC | ORCHECK, ORCSAVE2, ORKCHK |

| 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 |
| --- |
| RPC Call ‘ORWDXC ACCEPT’ kicks off order checking (including drug-allergy checks) which are returned |

| Modified Logic |
| --- |
|  |
| **Changes for NSR # 20071211**  **Added Function ALLERGY:**  ALLERGY(LST,DFN,FID,ORL,OIL) ; Return list of Order Checks on Accept Order  K ^TMP($J,"OROCOUTO;"),^TMP($J,"OROCOUTI;"),^TMP($J,"ORDSGCHK\_CACHE")  ; OIL(n)=OIptr^PS|PSIV|LR^PkgInfo  K ^TMP($J,"ORENHCHK"),^TMP($J,"ORALLERGYCHK")  N X,Y,USID,ORCHECK,ORI,ORX,ORY,%DT,ORDODSG  K ORX,ORY  ; do the ALLERGY order checks  S ORX(1)=+OIL\_"|"\_FID\_"||",(ORX,ORI)=1  D EN^ORKCHK(.ORY,DFN,.ORX,"ALLERGY",.OIL,0)  I $D(ORY) D RETURN^ORCHECK ; expects ORY, ORCHECK  ; return ORCHECK as 1 dimensional list  D FDBDOWN^ORCHECK(0)  I $D(ORY) M ^TMP($J,"ORALLERGYCHK")=ORY  D CHK2LST  K ^TMP($J,"OROCOUTO;"),^TMP($J,"OROCOUTI;"),^TMP($J,"DD"),^TMP($J,"ORDSGCHK\_CACHE")  Q  **Added Function REASON:**  REASON(LST,TYP,DFN,OID) ;Return list of pre-defined override reasons  N ORRSN,RSNI,RSNTYP,ORDT,ORVP,ORIFN,ORLAST  S ORDT="",ORIFN="",ORVP=DFN\_";DPT(",ORLAST=""  I OID D  . F S ORDT=$O(^OR(100,"AOI",OID,ORVP,ORDT),-1) Q:ORDT="" D  . . F S ORIFN=$O(^OR(100,"AOI",OID,ORVP,ORDT,ORIFN)) Q:ORIFN=""!(ORLAST]"") D  . . . Q:'$D(^ORD(100.05,ORIFN,3,1))  . . . S ORLAST=$G(^ORD(100.05,ORIFN,3,1,0))  S ORRSN=0,RSNI=0  I ORLAST]"" S LST($I(RSNI))=ORLAST  F S ORRSN=$O(^ORD(100.04,ORRSN)) Q:'ORRSN D  . S RSNTYP=$P(^ORD(100.04,ORRSN,0),"^",3)  . Q:RSNTYP'=TYP&(RSNTYP'="B")  . Q:$P(^ORD(100.04,ORRSN,0),"^",1)=ORLAST  . S LST($I(RSNI))=$P(^ORD(100.04,ORRSN,0),"^",1)  I 'RSNI S LST(1)="No predefined reasons available"  Q  Changes related to NSR # 20101203  Updated SAVECHK:  SAVECHK(OK,ORVP,LSTR,LSTC,LST) ; Save order checks for session  N ORCHECK,ORIFN,**ORREASONS,ORCOMMENTS** S OK=1  D LST2CHK  **;I $L(RSN)>0 S ORCHECK("OK")=RSN RSN parameter replaced by LSTR array)**  S ORIFN=0 F S ORIFN=$O(ORCHECK(ORIFN)) Q:'ORIFN D  **. S ORCHECK("OK")=""**  **. I $D(ORREASONS(ORIFN)) S $P(ORCHECK("OK"),U)=ORREASONS(ORIFN)**  **. I $D(ORCOMMENTS(ORIFN)) S $P(ORCHECK("OK"),U,2)=ORCOMMENTS(ORIFN)**  . D OC^ORCSAVE2  Q  Added in LST2CHK:  I $D(LSTR) D  . S ILST="" F S ILST=$O(LSTR("ORREASONS",ILST)) Q:$L(ILST)'>0 D  . . S ORIFN=+$P(LSTR("ORREASONS",ILST),U) S ORREASONS(ORIFN)=$P(LSTR("ORREASONS",ILST),U,2)  I $D(LSTC) D  . S ILST="" F S ILST=$O(LSTC("ORCOMMENTS",ILST)) Q:$L(ILST)'>0 D  . . S ORIFN=+$P(LSTC("ORCOMMENTS",ILST),U) S ORCOMMENTS(ORIFN)=$P(LSTC("ORCOMMENTS",ILST),U,2) |

###### ORKCHK

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORWDXC (ALLERGY and ACCEPT) | EN^ORKCHK |

| 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 |
| --- |
| “ACCEPT” call will trigger a call to EN^ORKCHK5 |

| Modified Logic (Changes are in bold) |
| --- |
| Updated line in EN^ORKCHK to add option for ALLERGY:  ..I ORKMODE="ACCEPT" ! (ORKMODE="ALLERGY") D EN^ORKCHK5(.ORKS,ORKDFN,ORKA(ORKX),ORENT,ORKTMODE,.OROIL,.ORDODSG) |

###### ORKCHK5

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORKCHK5 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| In tag EN:  D**:ORKMODE'="ALLERGY"** REMCHK(.ORKS,OI,ORKDFN) ;do reminder order checks  PHARM+8:9  .D**:ORKMODE="ALLERGY"** RXOCS  .D**:ORKMODE'="ALLERGY"** MLM^ORKCHK2(.ORKS,ORKDFN,ORKA,ORENT,"ACCEPT")  PHARM+24:25:  ..D**:ORKMODE="ALLERGY"** RXOCS  ..D**:ORKMODE'="ALLERGY"** MLM^ORKCHK2(.ORKS,ORKDFN,ORKA,ORENT,"ACCEPT")  RXOCS+4:5:  .N SIGN,GMRALLER,REACTANT,TEXT,ITM,ITEMS,NODE,COUNT,SEVERE,K,SITE**,ORREMOTE,ORREMLST**  .S FCOUNT=2,GMRALLER=$P(DATA(J,"MESSAGE",2),U,3),REACTANT=$P(DATA(J,"MESSAGE",2),U,2)**,ORREMOTE=0,ORREMLST=""**  RXOCS+41:43:  ..;check for remote allergies and related comments  ..I $P(DATA(J,"MESSAGE",1,ITM),U,2)["REMOTE" D  ... S ORREMOTE=1  RXOCS+59:  ..S ^TMP("OROCIDATA",$J,CRC16,100.517,NUM,11)=$P(DATA(J,ITM),U,10)  RXOCS+66:68: (Line 68 moved down from RXOCS+46)  ..S ORREMLST=$P(DATA(J,ITM),U)\_";"\_$P(DATA(J,ITM),U,6)\_"^"\_$P(DATA(J,ITM),U,10)  .;Now save the message to the ORKS array (moved here to allow for remote allergy comment changes)  .S ORKS("ORK",ORALLRD\_","\_$G(ORNUM)\_","\_$E(ORKMSG,1,225))=ORNUM\_U\_ORALLRN\_U\_ORALLRD\_U\_ORKMSG\_U\_ORREMOTE\_U\_ORREMLST |

###### ORCSAVE

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | SAVE^ORWDX | SAVEOC^OROCAPI1 |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Add ability to store an override reason in the existing order save call.  Added at EN2+33-34:  . . . s ORK(I,3)=$P(X,U,4)  . . . s ORK(I,4)=$P(X,U,5) |

###### ORCSAVE2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORCSAVE2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | SAVECHK^ORWDXC | SAVEOC^OROCAPI1 |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Add ability to store an override reason in the existing order save call.  Added at OC+12:  . . S ORK(CNT,4)=$S(I=1:$P($G(ORCHECK("OK")),U,2),1:"") |

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

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Add ability to store an override reason in the existing order save call.  Added into SAVE+3:  N ORDUZ,ORSTS,OREVENT,ORCAT,ORDA,ORTS,ORNEW,ORCHECK,ORLOG,ORLEAD,ORTRAIL,ORPKG,ORWP94,ORCATFN,OREVTYPE,ONPASS**,OROVER,ORAGYCNT,ORAGY,ORCHKCNT,ORRCOMM**  SAVE+19:20:  I $D(ORDIALOG("OVERRIDE")) S OROVER=ORDIALOG("OVERRIDE") K ORDIALOG("OVERRIDE")  I $D(ORDIALOG("ORREMCOMMENT")) S ORRCOMM=ORDIALOG("ORREMCOMMENT") K ORDIALOG("ORREMCOMMENT")  SAVE+47:55 :  I $D(^TMP($J,"ORALLERGYCHK")) D  . I $G(OROVER)="" S OROVER="No override reason given"  . S ORAGYCNT=$O(^TMP($J,"ORALLERGYCHK",""),-1),ORCHKCNT=$O(ORCHECK("NEW",1,""),-1),ORCHECK=ORCHECK+ORAGYCNT  . F ORAGY=1:1:ORAGYCNT D  . . S ORCHECK("NEW",1,$I(ORCHKCNT))=$P(^TMP($J,"ORALLERGYCHK",ORAGY),U,2,4)\_U\_$G(OROVER)  . . I $G(ORRCOMM)]"",$P(^TMP($J,"ORALLERGYCHK",ORAGY),U,5)=1 D  . . . S $P(ORCHECK("NEW",1,ORCHKCNT),U,5)=ORRCOMM  . . . D SAVRCOM(ORVP,$P(^TMP($J,"ORALLERGYCHK",ORAGY),U,6),ORRCOMM)  K ^TMP($J,"ORALLERGYCHK")  Added SAVRCOM:  SAVRCOM(ORVP,AREC,RCOMM) ;Save Local Comment to Remote Allergy  ;AREC: This will contain the allergy record identifier (RECID)  ; and the original comment presented to the user (PREVCOMM)  N GMR,RECID,PREVCOMM,COMREC,DA  S RECID=$P(AREC,"~"),PREVCOMM=$P(AREC,"~",2,99)  S DIC="^GMR(120.88,",DIC(0)="F"  S DIE=DIC  S COMMID=$O(^GMR(120.88,"PR",ORVP,RECID,""))  I COMMID="" S COMMID="+1,"  S DA(.01)=RECID,DA(.02)=ORVP,DA(.03)=$$NOW^XLFDT(),DA(.04)=DUZ,DA(1)=RCOMM  M GMR(120.88,COMMID)=DA  I PREVCOMM=RCOMM Q  D UPDATE^DIE("","GMR",,"ERROR")  Q |

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

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Updated RETURN+4:  . S ORCHECK=+$G(ORCHECK)+1,ORCHECK(IFN,CDL,ORCHECK)=$P(ORY(I),U,2,**7**) |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | OROCAPI1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Changes in SAVEOC  SAVEOC+10:  **; (D0,4)=COMMENT FOR A REMOTE ALLERGY**  SAVEOC+41  **...I $G(ORL(I,4))]"" S ^TMP("OROCIDATA",$J,CRC16,100.517,NUM,11)=ORL(I,4)** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAOR0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| Added in REMOTE:  REMOTE+72:  .S COMMID=$O(^GMR(120.88,"PR",DFN\_";DPT(",$P(DATA,U)\_";"\_$P(DATA,U,6),""))  .S $P(DATA,U,10)=$S(COMMID="":"",1:$G(^GMR(120.88,COMMID,1))) |

##### NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail

###### PSBOAL

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSBOAL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | EN1^GMRADPT  EN1^GMRAOR2 |

| 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 |
| --- |
| EN1 ;  S GMRA="0^0^111" S PSBLIST=""  D EN1^GMRADPT  F S PSBLIST=$O(GMRAL(PSBLIST)) Q:'PSBLIST D  .K PSBARRY  .D EN1^GMRAOR2(PSBLIST,"PSBARRY")  .W !,""  .W !," Causative agent : "\_$P(PSBARRY,U)  .W !,""  .I $D(PSBARRY("S",1)) D SYMP  .I $D(PSBARRY("V",1)) D CLAS  .W !," Verified : "\_$S($P(PSBARRY,U,4)="VERIFIED":"Yes",1:"No")  .W !," Observed/Historical : "\_$S($P(PSBARRY,U,5)="OBSERVED":"Observed",$P(PSBARRY,U,5)="HISTORICAL":"Historical",1:"")  .W !," \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  Q  ; |

| Modified Logic (Changes are in bold) |
| --- |
| EN1 ;  S GMRA=**"0^0^111^1"** S PSBLIST=""  D EN1^GMRADPT  F S PSBLIST=$O(GMRAL(PSBLIST)) Q:'PSBLIST D  .K PSBARRY  .D EN1^GMRAOR2(PSBLIST,"PSBARRY")  .W !,""  .W !," Causative agent : "\_$P(PSBARRY,U)  **.W !," Nature of Reaction : "\_$S($P(PSBARRY,U,6)="ALLERGY":"Allergy",$P(PSBARRY,U,6)="PHARMACOLOGIC":"Adverse Reaction",$P(PSBARRY,U,6)="UNKNOWN":"Unknown",1:"") ;93**  .W !,""  .I $D(PSBARRY("S",1)) D SYMP  .I $D(PSBARRY("V",1)) D CLAS  **.W !,**  **.W !," Originator : "\_$P(PSBARRY,U,2)\_$S($L($P(PSBARRY,U,3)):" ("\_$P(PSBARRY,U,3)\_")",1:"") ;93**  **.W !," Originated : "\_$P(PSBARRY,U,10) ;93**  **. I $D(PSBARRY("O",1)) D OBS**  .W !," Verified : "\_$S($P(PSBARRY,U,4)="VERIFIED":"Yes",1:"No")  .W !," Observed/Historical : "\_$S($P(PSBARRY,U,5)="OBSERVED":"Observed",$P(PSBARRY,U,5="HISTORICAL":"Historical",1:"")  **.I $D(PSBARRY("O",1)) D OBS**  **.I $D(PSBARRY("C",1)) D COM**  .W !," \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  Q  **OBS ;Display originator name ;added tag 93**  **N N,K,Y**  **S K=0,N=0 F S K=$O(PSBARRY("O",K)) Q:K'>0 D**  **.I N=0 D**  **..S Y=$P(PSBARRY("O",K),U) D DD^%DT**  **..W !," Date/Severity : "\_Y\_" "\_$P(PSBARRY("O",K),U,2)**  **.E D**  **..S Y=$P(PSBARRY("O",K),U) D DD^%DT**  **..W !," "\_Y\_" "\_$P(PSBARRY("O",K),U,2)**  **.S N=N+1**  **K N,K,Y**  **Q**  **COM ;display comments - tag added 93**  **N N,K,L,Y**  **S K=0,N=0,Y=0**  **F S K=$O(PSBARRY("C",K)) Q:K'>0 D**  **.I N=0 W !," Comments :"**  **.S Y=$P(PSBARRY("C",K),U) D DD^%DT**  **.W " "\_Y\_" by "\_$P(PSBARRY("C",K),U,2)**  **.I $D(PSBARRY("C",K,1,0)) S L=0 F S L=$O(PSBARRY("C",K,L)) Q:L'>0 D**  **.. W !," ",PSBARRY("C",K,L,0)**  **.S N=N+1**  **W !**  **K N,K,L,Y**  **Q** |

###### PSBALL

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSBALL | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | PSBOHDR | EN2^GMRADPT |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** |  | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** | EN2^GMRADPT/10099  HAVEHDR^ORRDI1 | | | | |
| **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 |
| --- |
| SORT ;\*\*\* Set up the allergies and adv. reactions arrays.  S PSBCNT=1,(PSBALC,PSBARC)=0  S GMRA="0^0^111" D EN1^GMRADPT  I GMRAL="" S RESULTS(0)=1,RESULTS(1)="ALL"\_U\_"No Allergy Assessment" Q  I GMRAL=0 S RESULTS(0)=1,RESULTS(1)="ALL"\_U\_"No Known Allergies" Q  F X=0:0 S X=$O(GMRAL(X)) Q:'X S PSBTYP=$P(GMRAL(X),U,5),PSBNM=$P(GMRAL(X),U,2) D  .S:PSBTYP=0 PSBALC=PSBALC+1,PSBALG(PSBNM)=""  .S:PSBTYP>0 PSBARC=PSBARC+1,PSBADR(PSBNM)=""  S (X,Y)="" F S X=$O(PSBALG(X)) Q:X="" S Y="ALL"\_U\_X,RESULTS(PSBCNT)=Y,PSBCNT=PSBCNT+1  S (X,Y)="" F S X=$O(PSBADR(X)) Q:X="" S Y="ADR"\_U\_X,RESULTS(PSBCNT)=Y,PSBCNT=PSBCNT+1  S RESULTS(0)=PSBCNT-1  K GMRA,GMRAL,PSBADR,PSBALC,PSBALG,PSBARC,PSBCNT,PSBNM,PSBTYP  Q  ; |

| Modified Logic (Changes are in bold) |
| --- |
| SORT ;\*\*\* Set up the allergies and adv. reactions arrays.  **S PSBCNT=1,(PSBALC,PSBARC,PSBREM)=0,(W,X,ALLADR)="" N REMOTE ;93**  **;S GMRA="0^0^111" D EN1^GMRADPT ;93**  **S GMRA="0^0^111^1" D EN2^GMRADPT ;93 INCLUDE REMOTE ALLERGIES**  I GMRAL="" S RESULTS(0)=1,RESULTS(1)="ALL"\_U\_"No Allergy Assessment" Q  I GMRAL=0 S RESULTS(0)=1,RESULTS(1)="ALL"\_U\_"No Known Allergies" Q  **F S X=$O(GMRAL(X)) Q:X="" D ;93**  **.S PSBTYP=$P(GMRAL(X),U,5),PSBNM=$P(GMRAL(X),U,2)**  **.I X["R" D Q ;93 check if there were remote allergies returned**  **..S PSBREM=PSBREM+1,PSBREMC(PSBNM)="" ;93**  **.S:PSBTYP=0 PSBALC=PSBALC+1,PSBALG(PSBNM)=""**  **.S:PSBTYP>0 PSBARC=PSBARC+1,PSBADR(PSBNM)=""**  S (X,Y)="" F S X=$O(PSBALG(X)) Q:X="" S Y="ALL"\_U\_X,RESULTS(PSBCNT)=Y,PSBCNT=PSBCNT+1  S (X,Y)="" F S X=$O(PSBADR(X)) Q:X="" S Y="ADR"\_U\_X,RESULTS(PSBCNT)=Y,PSBCNT=PSBCNT+1  **I $D(PSBREMC) D ;\*93**  **.S X="",X=$O(PSBREMC(X))**  **.S W=PSBCNT-1,ALLADR=$P(RESULTS(W),U)**  **.S RESULTS(W)=RESULTS(W)\_" OTHER FACILITIES: "\_X**  **.S Y="" F S X=$O(PSBREMC(X)) Q:X="" S Y=ALLADR\_U\_X,RESULTS(PSBCNT)=Y,PSBCNT=PSBCNT+1**  **S REMOTE=$$HAVEHDR^ORRDI1 ;check if server is up**  **I REMOTE'>0 S W=PSBCNT-1,ALLADR=$P(RESULTS(W),U),RESULTS(W)=RESULTS(W)\_" OTHER FACILITIES: Remote checking unavailable"**  S RESULTS(0)=PSBCNT-1  K ALLADR,GMRA,GMRAL,PSBADR,PSBALC,PSBALG,PSBARC,PSBCNT,PSBNM,PSBTYP,**PSBREM,PSBREMC,W,X,Y** ;93  Q |

###### PSODO

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSODO | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | PSBOHDR  PBBO  PSBVT  PSGSICH1  PSJBCMA2  PSJBCMA5  XLFDT  DIQ |

| 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 |
| --- |
| DISPORD ;  .  .  .;\*68 change  .W !,"Special Instructions/Other Print Info:"  .K ^TMP("PSJBCMA5",$J)  .D GETSIOPI^PSJBCMA5(DFN,PSBONX,1)  .F QQ=0:0 S QQ=$O(^TMP("PSJBCMA5",$J,DFN,PSBONX,QQ)) Q:'QQ D  ..W !,^TMP("PSJBCMA5",$J,DFN,PSBONX,QQ)  .K ^TMP("PSJBCMA5",$J)  .;\*68 end  .;\*58 override/intervention section \* \* \*  .S PSBOVR=0  .D GETPROVL^PSGSICH1(DFN,PSBONX,.PSBPRV)  .D INTRDIC^PSGSICH1(DFN,PSBONX,.PSBRPH,2)  .S PSBPV=$S($D(PSBPRV)>1:1,1:0)  .S PSBRH=$S($D(PSBRPH)>1:1,1:0)  .I 'PSBPV,PSBRH D DSPPRV(.PSBPRV,132,2,26,1) S PSBOVR=1  .I PSBPV D DSPPRV(.PSBPRV,132,2,26) S PSBOVR=1  .I PSBPV,'PSBRH D DSPRPH(.PSBRPH,132,2,26,1) S PSBOVR=1  .I PSBRH D DSPRPH(.PSBRPH,132,2,26) S PSBOVR=1  .I PSBOVR W !,$TR($J("",75)," ","-")  .;\*58 end override/intervention section \* \* \*  .;  ...W !  .  .  W !!  D CLEAN^PSBVT K @(PSJGLO\_")")  Q |

| Modified Logic (Changes are in bold) |
| --- |
| **DISPORD ; THE FOLLOWING SECTION IS ADDED TO DISPORD**  **.;\*93 begin**  **.S (PSBORCK,CNT)=0**  **.N ORCPRS,ORIFN,LST,ORY**  **.K ^TMP("ORTXT",$J)**  **.S ORCPRS=$S(PSBONX["U":$P($G(^PS(55,DFN,5,+PSBONX,0)),U,21),PSBONX["V":$P($G(^PS(55,DFN,"IV",+PSBONX,0)),U,21),PSBONX["P":$P($G(^PS(53.1,+PSBONX,0)),U,21),1:0)**  **.S ORIFN=ORCPRS,LST=$NA(^TMP("ORTXT",$J)),@LST="",ORY=LST**  **.D D5^ORQ2**  **.M DSPORCK("ORTXT",$J)=^TMP("ORTXT",$J)**  **.S ORCKFL=$S($D(DSPORCK("ORTXT",$J))>1:1,1:0)**  **.I 'ORCKFL D DSPORCK(.DSPORCK,132,2,26,1)**  **.I ORCKFL D DSPORCK(.DSPORCK,132,2,26)**  **.K ^TMP("ORTXT",$J)**  **.;\*93 end**  **DSPORCK(ARR,LN,IND,ALGN,NONE) ; Display Order Check(CPRS) #93 NEW TAG**  **; ARR = array with order check text.**  **; LN = total width of report writable area. (opt, 132 default)**  **; IND = indent for both left and right margins. (opt,0 default)**  **; ALGN = align colon on this column. (opt, 25 default)**  **; NONE = display empty order check msg. (opt, 0 default)**  **;**  **N CAT,QQ,OC,HDG,CTRTAB**  **S LN=+$G(LN,132),IND=+$G(IND),ALGN=$G(ALGN,25),NONE=$G(NONE,0)**  **S LN=LN-(IND\*2) ;adj writeable area by both L & R margins**  **;order check heading**  **W !!?IND,$TR($J("",LN)," ","=")**  **S HDG="\*\* Current Order Check \*\*"**  **S CTRTAB=(LN-$L(HDG))/2**  **W !?CTRTAB,HDG**  **W !?IND,$TR($J("",LN)," ","="),!**  **;**  **; if there isn't order check information to display**  **I NONE W !?IND,"No Order Check to display.",! Q**  **;**  **;write out order check text**  **S QQ=0 F S QQ=$O(^TMP("ORTXT",$J,QQ)) Q:QQ'>0 W ?IND,$G(^TMP("ORTXT",$J,QQ)),!**  **Q** |

###### PSODO

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSODO (cont.) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR #20100825 - Drug-Allergy Order Check Enhancements and Improved Detail | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | PSBOHDR  PBBO  PSBVT  PSGSICH1  PSJBCMA2  PSJBCMA5  XLFDT  DIQ |

| 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 |
| --- |
| New code added |

| Modified Logic (Changes are in bold) |
| --- |
| **ORCHECK ; recreate CPRS Order Check - copied from ORQ2**  **K ^TMP($J,"OCDATA")**  **I '$$OCAPI^ORCHECK(+ORIFN,"OCDATA") D DSPORCK(.DSPORCK,132,2,26,1) Q**  **E D**  **. N CK,OK,X0,X,CDL,I,ACK,ALLGYDRG,HDR S HDR=0**  **. S ACK=0**  **. D ALLERGY ;405**  **. S:$D(OK) OK="" ;405**  **. S CK=0 F S CK=$O(^TMP($J,"OCDATA",CK)) Q:CK'>0 D**  **.. Q:$D(ALLGYDRG(CK)) ;skip allergy entries**  **.. S:HDR=0 CNT=CNT+1,@ORY@(CNT)=" ",CNT=CNT+1,@ORY@(CNT)="Order Checks:",CNT=CNT+1,@ORY@(CNT)=" ",HDR=1**  **.. S X0=^TMP($J,"OCDATA",CK,"OC NUMBER")\_U\_^TMP($J,"OCDATA",CK,"OC LEVEL")\_U\_U\_^TMP($J,"OCDATA",CK,"OR REASON")\_U\_^TMP($J,"OCDATA",CK,"OR PROVIDER")\_U\_^TMP($J,"OCDATA",CK,"OR DT")**  **.. S X=^TMP($J,"OCDATA",CK,"OC TEXT",1,0)**  **.. S CDL=$$CDL($P(X0,U,2)) I $P(X0,U,6),'$D(OK) S OK=$P(X0,U,4,6)**  **.. I $L(X)'>100 S CNT=CNT+1,@ORY@(CNT)=CDL\_X D XTRA Q**  **.. S DIWL=1,DIWR=100,DIWF="C100" K ^UTILITY($J,"W") D ^DIWP ;change line length to 100**  **.. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=CDL\_^(I,0),CDL=" "**  **.. D XTRA**  **. K ^TMP($J,"OCDATA")**  **. Q:(HDR=0) ;405 quit if only allergy-drug interactions**  **. Q:'$L($G(OK)) S CNT=CNT+1,@ORY@(CNT)="Override: "\_$S($P(OK,U,2):$$USER^ORQ20($P(OK,U,2))\_" on ",1:"")\_$$DATE^ORQ20($P(OK,U,3))**  **. I $L($P(OK,U))'>100 S CNT=CNT+1,@ORY@(CNT)=" "\_$P(OK,U) Q**  **. S DIWL=1,DIWR=100,DIWF="C100",X=$P(OK,U) K ^UTILITY($J,"W") D ^DIWP**  **. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=" "\_^(I,0)**  **D DSPORCK(.DSPORCK,132,2,26) ;print order check**  **K ^TMP("ORWORD",$J),^UTILITY($J,"W"),ALLGYDRG**  **Q**  ;  **ALLERGY ;separate the ALLERGY-DRUG INTERACTION Order Checks (405)**  **N RET,INSTANCE,INSTASAV**  **S RET=1,(ACK,CK,CNT)=0**  **F S CK=$O(^TMP($J,"OCDATA",CK)) Q:CK'>0 D**  **. I $G(^TMP($J,"OCDATA",CK,"OC NUMBER"))'=3 Q ;only select the allergy-drug interactions**  **. S ALLGYDRG(CK)=" "**  **. I ACK=0 S CNT=CNT+1,@ORY@(CNT)=" ",CNT=CNT+1,@ORY@(CNT)="Allergy Order Checks:",CNT=CNT+1,ACK=1**  **. S @ORY@(CNT)=" ",CNT=CNT+1**  **. S X0=^TMP($J,"OCDATA",CK,"OC NUMBER")\_U\_^TMP($J,"OCDATA",CK,"OC LEVEL")\_U\_U\_^TMP($J,"OCDATA",CK,"OR REASON")\_U\_^TMP($J,"OCDATA",CK,"OR PROVIDER")\_U\_^TMP($J,"OCDATA",CK,"OR DT")**  **. S X=^TMP($J,"OCDATA",CK,"OC TEXT",1,0)**  **. S CDL=$$CDL($P(X0,U,2)) I $P(X0,U,6),'$D(OK) S OK=$P(X0,U,4,6)**  **. I $L(X)'>100 S CNT=CNT+1,@ORY@(CNT)=CDL\_X D XTRA Q**  **. S DIWL=1,DIWR=100,DIWF="C100" K ^UTILITY($J,"W") D ^DIWP**  **. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=CDL\_^(I,0),CDL=" "**  **. S INSTANCE=$G(^TMP($J,"OCDATA",CK,"OC INSTANCE")) ;NEW**  **. I INSTANCE>0 D**  **.. I $$GET1^DIQ(100.517,RET\_","\_INSTANCE\_",",11)'="" D ;NEW**  **... S:'$D(INSTASAV) INSTASAV=INSTANCE**  **... ;S CNT=CNT+1,@ORY@(CNT)="Remote Comment: "\_$$GET1^DIQ(100.517,RET\_","\_INSTANCE\_",",11) ;NEW**  **... ;S CNT=CNT+1,@ORY@(CNT)=" "**  **. D XTRA**  **I ACK=1 D ;if there are allergy-drug interaction check for override?**  **. Q:'$L($G(OK)) S CNT=CNT+1,@ORY@(CNT)="Override: "\_$S($P(OK,U,2):$$USER^ORQ20($P(OK,U,2))\_" on ",1:"")\_$$DATE^ORQ20($P(OK,U,3))**  **. I $L($P(OK,U))'>100 S CNT=CNT+1,@ORY@(CNT)=" "\_$P(OK,U),CNT=CNT+1 Q**  **. S DIWL=1,DIWR=100,DIWF="C100",X=$P(OK,U) K ^UTILITY($J,"W") D ^DIWP**  **. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=" "\_^(I,0)**  **I $D(INSTASAV) D**  **. S CNT=CNT+1,@ORY@(CNT)=" "**  **. S CNT=CNT+1,@ORY@(CNT)="Remote Comment: "\_$$GET1^DIQ(100.517,RET\_","\_INSTASAV\_",",11)**  **Q** |

##### NSR 20071103 – Change in Unflagging Capabilities

###### ORCACT0

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORCACT0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20071103 Change in Unflagging Capabilities | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |

| Modified Logic |
| --- |
| **Updates to UF (in Bold):**  UF I ACTION="UF" D G VQ ; unflag  . I '+$G(^OR(100,+IFN,8,AIFN,3)) S ERROR="This order is not flagged!" Q  **. Q:$D(^XUSEC("ORES",DUZ)) ; No restrictions if user holds ORES key**  **. Q:'$$GET^XPAR("DIV^SYS^PKG","OR UNFLAGGING RESTRICTIONS",1) ; No unflagging restrictions found**  **. ; Check Security Key multiple in Display Group file and compare with user**  **. N DGP,DGSK,ORSKP,SFND S DGP=$P(OR0,U,11),SFND=0**  **. I $D(^ORD(100.98,DGP,2)) D**  **. . S DGSK=0 F S DGSK=$O(^ORD(100.98,DGP,2,DGSK)) Q:DGSK=""!(DGSK'?1N.N) I $D(^ORD(100.98,DGP,2,DGSK,0)) D**  **. . . S ORSKP=^ORD(100.98,DGP,2,DGSK,0)**  **. . . I $D(^XUSEC($$GET1^DIQ(19.1,ORSKP\_",",.01,"E"),DUZ)) S SFND=1**  **. ; If user doesn't hold proper security key(s), send site-customizable help message**  **. I 'SFND S ERROR=$$GET^XPAR("DIV^SYS^PKG","OR UNFLAGGING MESSAGE",1) Q** |

##### NSR 20110210 – Create Separate Alert for Prosthetics Requests Screening Call Record

###### GMRCP

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | MSG^GMRCP | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | 2.6.27, 2.6.27.1 | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | GMRCAAC, GMRCACMT, GMRCADC, GMRCAFRD,  GMRCCP, GMRCDIS,  GMRCGUIA, GMRCGUIB,  GMRCGUIC, GMRCGUIS,  GMRCHL7B, GMRCHL7I,  GMRCHL7U, GMRCIAC1,  GMRCIACT, GMRCMED,  GMRCTIU1 | EN^ORB3  EN^GMRCT |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** | 1362 – Consult calls to EN^ORB3 | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | Local |
| **Input Attribute Name and Definition** | Name: GMRCDFN  Definition: Patient Internal Entry Number  Name: GMRCALRM  Definition: Alert message displayed with alert  Name: GMRCIFN  Definition: Internal Entry Number of consult from Consult/Request Tracking (#123)  Name: ORN  Definition: Internal Entry Number of notification from OE/RR Notifications (#100.9)  Name: GMRCADUZ  Definition: Array of all providers who would receive notification  Name: FLG  Definition: Local variable to control how GMRCADUZ is populated | | | | |
| **Output Attribute Name and Definition** | Name: ORN  Definition: Internal Entry Number of notification from OE/RR Notifications (#100.9)  Name: GMRCDFN  Definition: Patient Internal Entry Number  Name: GMRCORFN  Definition: Internal Entry Number of order from Orders (#100)  Name: GMRCADUZ  Definition: Array of all providers who would receive notification  Name: GMRCALRM  Definition: Alert message displayed with alert  Name: GMRCIFN  Definition: Internal Entry Number of consult from Consult/Request Tracking (#123) | | | | |

| Current Logic |
| --- |
| …  K GMRCADUZ(DUZ) ;Don't send alert to user generating alert  D EN^ORB3(ORN,GMRCDFN,GMRCORFN,.GMRCADUZ,GMRCALRM,GMRCIFN)  Q |

| Modified Logic (changes in bold) |
| --- |
| …  K GMRCADUZ(DUZ) ;Don't send alert to user generating alert  **I $G(^GMR(123.5,GMRCSS,"INT"))=1 D ;set ORN to prosthetics alert, if necessary**  **. S:$G(ORN)=63 ORN=89**  D EN^ORB3(ORN,GMRCDFN,GMRCORFN,.GMRCADUZ,GMRCALRM,GMRCIFN)  Q |

###### GMRAGUI1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAGUI1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | RSD #2.6.13.4.1 | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

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

###### ORWDAL32

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDAL32 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | RSD #2.6.13.4 | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

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

###### ORWDX2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDX2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

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

###### PSJHL5

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSJHL5 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | PSJHL4, PSJHL4A | %DTC, ALPBCBU, DIE, ORERR, ORHLESC, PSGAL5, PSIVORAL, PSJHL2, PSJHL6, PSJHLERR, PSJHLV, PSJLOI |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | PS RECEIVE OR, ORC VERIFY PROBLEM | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **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) |
| --- |
|  |

###### ORDV06

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORDV06 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | CPRS Reports (GUI): Active Outpatient, Outpatient Medications, Active IV, All IV, Unit Dose, Outpatient Medications, Pharmacy All Outpatient, HerbalOTC/Non-VA Meds, All Medications | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORWRP | PSJEEU0, OMGCOAS1, ODRDVU, VASITE |

| Routines | Activities | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | | | |
| **Related Protocols** | N/A | | | | | | |
| **Related Integration Control Registrations (ICRs)** | 486, 3486 | | | | | | |
| **Data Passing** | Input | | Output Reference | Both | | Global Reference | Local |
|  |  | This routine is invoked by going into the CPRS GUI on the Reports tab and selecting one of the reports above. | | |  |  |  |
| **Input Attribute Name and Definition** | Name: ORALPHA  Definition: Beginning date for search in $H format  Name: OROMEGA  Definition: Ending date for search in $H format  Name: ORMAX  Definition: Maximum number of records to return  Name: ORDBEG  Definition: Beginning date in FileMan format  Name: ORDEND  Definition: Ending date in FileMan format  Name: OREXT  Definition: Report definition information: type;routine;name | | | | | | |
| **Output Attribute Name and Definition** | Name: TMP  Definition: TMP global that contains a list of the orders that meet the user-specified criteria, along with the pertinent data for display in the GUI | | | | | | |

| Current Logic |
| --- |
|  |

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

###### ORQ21

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORQ21 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | CPRS Reports (GUI) | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORQ2, ORWDBA7 | DIQ, DIWP, ICDXCODE, ORWDBA3, PSOORRL, RAO7PC1, XLFDT, XLFSTR |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | N/A | | | | |
| **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) |
|  |

###### ORWDPS1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDPS1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | CPRS Outpatient Order Dialog, AudioCare Auto Renewal | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORALWORD, ORAREN, ORCACT01, ORCMEDT8, ORCSAVE, ORWDPS1, ORWDXR, ORWORR | PSSOPKI1, DIQ, ORCD, ORWDPS32, PSJORUTL, PSOSIGDS, PSS51P1, PSSDIN, PSSJORDF, PSSOPKI, PSSOPKI1, PSSORUTL, PSSUTIL1, XLFSTR, XPAR, XUSER |

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

| Current Logic |
| --- |
|  |

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

###### TIUSRVL0

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | TIUSRVL0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |
|  |
|  |
|  |
|  |

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

###### ORBPRCHK

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORBPRCHK | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | RSD #2.6.14.12.1, 2.6.14.2.2, 2.6.14.2.3 | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
|  |

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

##### NSR# 20090509 (15) – Park a Prescription

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORDV06,ORQ21,ORWDPS1,ORWDXA,ORWDXR,ORWPS1,PSOARX,PSOATRFC,PSOBGMG3,PSOCAN3,PSOCMOPA,PSODDPR5,PSODIR2,PSODIR3,PSODISP,PSODISP1,PSODRDU1,PSODRDU2,PSODSPL,PSOFUNC,PSOHCSUM,PSOHLDA,PSOLBL,PSOLLL7,PSOLMLST,PSOLMUTL,PSON52,PSONEW2,PSOORAL,PSOORAL1,PSOORDER,PSOORED1,PSOORED2,PSOORED6,PSOOREDT,PSOORFI1,PSOORFIN,PSOORNE1,PSOORNE2,PSOORNE4,PSOORNEW,PSOORRL,PSOORRL3,PSOORRLN,PSOORRLO,PSOORUT1,PSOP1,PSOPOLY,PSOPPAP1,PSOPRK,PSOPRKA,PSOQUAP2,PSOREF0,PSORENW,PSORENW3,PSORENW4,PSORLST2,PSORN52,PSORN52A,PSORPC01,PSORPTP,PSORXDL,PSORXED,PSORXED1,PSORXL,PSORXPA1,PSORXPR,PSORXRP1,PSORXRPT,PSORXVW,PSORXVW1,PSOSD0,PSOTRLBL,PSSDDUT2,PSSORUTE,TIULMED | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| 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 |
| --- |
| Outpatient Pharmacy and supporting applications do not allow ‘Parked’ prescriptions. ‘Park’ is not an option for a prescription’s status/state. Currently the pharmacy application only allows for two different routings for active prescriptions: mail, or window.  When window is selected, the prescription is picked up a the window, when mail is selected the prescription is sent to the patient via mail.  This also means that when a prescription is written and is active it has to be dispensed and the patient is not allowed to fill it at a later time. |

| Modified Logic |
| --- |
| A third method of routing will be added to Outpatient Pharmacy and supporting applications, Park(ed).  The Pharmacy application’s routines will be modified to allow prescription orders to be ‘Parked’ and ‘Un-parked’ where applicable, this will allow patients to fill active prescriptions at a later time.  The pharmacy related reports will now include the Parked status for orders that have parked prescriptions. Users will be able to change this status on active and inactive orders, for example, for deceased patients the system will allow this status to be modified. Code modified to not allow the medication Clozapine to be ‘Parked”. It checks the medications name for ‘Clozapine’ and will not present the ‘PARK’ option in VistA and will cause a pop up form stating that the medication cannot be ‘Parked’ in the GUI. ‘Parked’ medication will not generation an ECME claim. |

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Remote Procedure call “ORWORR GET4LST” |  |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** |  | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** | 4902 (accessing new tag PARK in PSO524X) | | | | |
| **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 |
| --- |
| GETFLDS ; used by entry points to place order fields into list  ; expects IDX=sequence #, IFN=order, X0=node 0, X3=node 3, LST=results  ; LST(IDX)=~IFN^Grp^OrdTm^StrtTm^StopTm^Sts^Sig^Nrs^Clk^PrvID^PrvNam^Act^Flagged[^DCType]^ChartRev^DEA#^^DigSig^LOC^[DCORIGNAL]^IsPendingDCorder^IsDelayOrder^IsParked  S PRV=$P(X8,U,5) S:'PRV PRV=$P(X8,U,3) S PRV=PRV\_U  I PRV S PRV=PRV\_$P(^VA(200,+PRV,0),U)  S DEA=$$DEA^XUSER(,+PRV) ; get user DEA info - PKI  S IDX=IDX+1,LST=LST+1,ID=IFN\_";"\_ACT,ACTID=$P(X8,U,2)  S CSTS=$S($P(X8,U,15):$P(X8,U,15),1:$P(X3,U,3))  I $P(X8,U,15)=10,$P(X3,U,3)=14 S CSTS=14 ;delayed-lapsed order  S STRT=$S($P(X3,U,3)=11:$$RSTRT,ACTID="NW"!(ACTID="XX")!(ACTID="RL"):$P(X0,U,8),ACTID="DC":"",1:$P(X8,U)) ;110  S STOP=$S($P(X3,U,3)=11:$$RSTOP,ACTID="HD":$P($G(^OR(100,+IFN,8,ACT,2)),U),1:$P(X0,U,9))  S LST(IDX)="~"\_ID\_U\_$P(X0,U,11)\_U\_$P(X8,U)\_U\_STRT\_U\_STOP\_U\_CSTS\_U\_$P(X8,U,4)\_U\_$P(X8,U,8)\_U\_$P(X8,U,10)\_U\_PRV  S $P(LST(IDX),U,13)=+$G(^OR(100,IFN,8,ACT,3)) ; flagged |

| Modified Logic (Changes are in bold) |
| --- |
| GETFLDS ; used by entry points to place order fields into list  ; expects IDX=sequence #, IFN=order, X0=node 0, X3=node 3, LST=results  ; LST(IDX)=~IFN^Grp^OrdTm^StrtTm^StopTm^Sts^Sig^Nrs^Clk^PrvID^PrvNam^Act^Flagged[^DCType]^ChartRev^DEA#^^DigSig^LOC^[DCORIGNAL]^IsPendingDCorder^IsDelayOrder^IsParked  S PRV=$P(X8,U,5) S:'PRV PRV=$P(X8,U,3) S PRV=PRV\_U  I PRV S PRV=PRV\_$P(^VA(200,+PRV,0),U)  S DEA=$$DEA^XUSER(,+PRV) ; get user DEA info - PKI  S IDX=IDX+1,LST=LST+1,ID=IFN\_";"\_ACT,ACTID=$P(X8,U,2)  S CSTS=$S($P(X8,U,15):$P(X8,U,15),1:$P(X3,U,3))  I $P(X8,U,15)=10,$P(X3,U,3)=14 S CSTS=14 ;delayed-lapsed order  S STRT=$S($P(X3,U,3)=11:$$RSTRT,ACTID="NW"!(ACTID="XX")!(ACTID="RL"):$P(X0,U,8),ACTID="DC":"",1:$P(X8,U)) ;110  S STOP=$S($P(X3,U,3)=11:$$RSTOP,ACTID="HD":$P($G(^OR(100,+IFN,8,ACT,2)),U),1:$P(X0,U,9))  S LST(IDX)="~"\_ID\_U\_$P(X0,U,11)\_U\_$P(X8,U)\_U\_STRT\_U\_STOP\_U\_CSTS\_U\_$P(X8,U,4)\_U\_$P(X8,U,8)\_U\_$P(X8,U,10)\_U\_PRV  **I $$PARK^PSO52EX(+IFN) S $P(LST(IDX),U,25)="active/park" ;405 set status of prescription if it is "PARKED"**  S $P(LST(IDX),U,13)=+$G(^OR(100,IFN,8,ACT,3)) ; flagged |

| **Routines** | **Activities** | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWPS1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| **Related Routines** | **Routines “Called By”** | **Routines “Called”** |
| --- | --- | --- |
|  | Remote Procedure Call: ORWPS1 REFILL |  |

| **Routines** | **Activities** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** |  | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** | 4902 (tag PARK in PSO52EX) | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | Local |
| **Input Attribute Name and Definition** | Name: ORDERID  Definition: IEN to the ORDER file (#100) | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| **Current Logic** |
| --- |
| REFILL(Y,ORDERID,REFLOC,ORVP,ORNP,ORL) ; Refill Request  S ORVP=ORVP\_";DPT(",ORL(2)=ORL\_";SC(",ORL=ORL(2)  D REF^ORMBLDPS(ORDERID,REFLOC)  S Y=""  Q |

| **Modified Logic (Changes are in bold)** |
| --- |
| REFILL(Y,ORDERID,REFLOC,ORVP,ORNP,ORL) ; Refill Request  **D CHKPARK I UNPARK S Y="" Q ;ADDED LINE OF PAPI CODE**  S ORVP=ORVP\_";DPT(",ORL(2)=ORL\_";SC(",ORL=ORL(2)  D REF^ORMBLDPS(ORDERID,REFLOC)  S Y=""  Q  **CHKPARK ; IF ORDER IS PARKED AND LAST FILL HAS NOT PRINTED A LABEL, REUSE LAST FILL INSTEAD OF ORDERING A NEW REFILL - 405**  **N ERRMSG**  **S UNPARK=0**  **I '$$PARK^PSO52EX(+ORDERID) Q ;ICR 4902**  **S PSODA=$O(^PSRX("APL",+ORDERID,""))**  **D UNPARK^PSOPRKA(PSODA,$P(ORVP,";"),.ERRMSG)**  **I $G(ERRMSG(1))'="" Q**  **S UNPARK=1**  **Q** |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSO52EX | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

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

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^PSRX(“APL”)  ^PSRX(“PARK”)  ^PSRX(“STA”) | | | | |
| **Related Protocols** |  | | | | |
| **Related Integration Control Registrations (ICRs)** | 4902 (adding PARK tag) | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | Local |
| **Input Attribute Name and Definition** | Name: ORIEN  Definition: IEN to the ORDER file (#100) | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| Current Logic |
| --- |
| New tag added. |

| Modified Logic (Changes are in bold) |
| --- |
| **PARK(ORIEN) ; DETERMINES IF A PRESCRIPTION IS PARK**  **;ORIEN: ORDER FILE (#100) INTERNAL ENTRY NUMBER**  **;**  **I $G(ORIEN)="" Q 0**  **N PSOIEN**  **S PSOIEN=$O(^PSRX("APL",ORIEN,""))**  **I +PSOIEN=0 Q 0**  **I $G(^PSRX(PSOIEN,"STA"))'=0 Q 0**  **N RESULT S RESULT=0**  **I $G(^PSRX(PSOIEN,"PARK"))=1 S RESULT=1**  **Q RESULT** |

**NSR 20110606 – Similar Provider Names**

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWU1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20110606 – 11.3 (Similar Provider Names) | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORWTPN, ORWU | NAMEFMT^XLFNAME, UP^XLFSTR, GET^XUA4A72, PROVIDER^XUSER, LIST^[DIC](http://code.osehra.org/dox/Routine_DIC.html), GET1^[DIQ](http://code.osehra.org/dox/Routine_DIQ.html), ACTLOC^[ORWU](http://code.osehra.org/dox/Routine_ORWU.html), ALL^[VASITE](http://code.osehra.org/dox/Routine_VASITE.html), SCRDFCS^[TIULA3](http://code.osehra.org/dox/Routine_TIULA3.html) |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(3.1), ^DIC(49), ^SC(IEN), ^SC(“B”), ^VA(200), ^XSEC | | | | |
| **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 |
| --- |
| Added the Similar Provider functionality to the ‘Select a Provider….’ option. When selecting a Provider the Similar Provider window will appear if there are multiple providers with the first two letters of the last and first name. Only those Active Providers with access to CPRS Tab and CPRS Charts will be displayed. |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWU2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** | NSR 20110606 – 11.3 (Similar Provider Names) | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | COSIGNER RPC Call | $$NAMEFMT^XLFNAME $$PROVIDER^XUSER  $$ISA^USRLM  $$ALL^VASITE  $$REQCOSIG^TIULP  LIST^DIC  $$LKOPT^XPDMENU  $$ACCESS^XQCHK  $$FIND1^DIC |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^DIC(3.1), ^DIC(49), ^VA(200), ^TIU(8925) | | | | |
| **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 |
| --- |
| Added the Similar Provider functionality to the Cosigner option. When selecting a Provider the Similar Provider window will appear if there are multiple providers with the first two letters of the last and first name. Only those Active Providers with access to CPRS Tab and CPRS Charts will be displayed. |

##### NSR# 20080307 CWAD Post Auto-Demotion Rules Request (RSD #2.6.4)

###### Routine TIUCWD

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | ^DIE, $$CHKSUM^TIULC, AUDIT^TIUEDI1 |

| 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: TIUFROM  Definition: This is the Starting From postion to find notes for demotion | | | | |
| **Input Attribute Name and Definition** | Name: TIUTO  Definition: This is the Ending at position to find notes for demotion | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **TIUCWD ;SLC/TDP - TIU CWAD POSTING AUTO-DEMOTION ;01/13/15 12:19**  **;;1.0;TEXT INTEGRATION UTILITIES;\*\*291\*\*;Jun 20, 1997;Build 4**  **;**  **SILENT(TIUFROM,TIUTO) ;**  **I +$G(TIUDA)&('+$G(DFN)) D ;TO GET DFN BASED ON TIUDA IF NOT PRESENT**  **. S DFN=+$P($G(^TIU(8925,TIUDA,0)),"^",2)**  **Q:'$G(DFN)**  **N TIUNDT,TIUNIEN,TIUCLIK**  **Q:$G(^TIU(8925.1,+TIUFROM,0))']""**  **Q:$G(^TIU(8925.1,+TIUTO,0))']""**  **S (TIUNDT,TIUCLIK)=0 F S TIUNDT=$O(^TIU(8925,"APT",DFN,TIUFROM,7,TIUNDT)) Q:'+TIUNDT D**  **. S TIUNIEN=0 F S TIUNIEN=$O(^TIU(8925,"APT",DFN,TIUFROM,7,TIUNDT,TIUNIEN)) Q:'+TIUNIEN D**  **.. S TIUCLIK=TIUCLIK+1 Q:(TIUCLIK'>1) ;LEAVE THE MOST RECENT NOTE**  **.. D DEMOTE(TIUNIEN,TIUTO)**  **Q**  **DEMOTE(TIUDA,TIUTO) ;**  **N DIE,DR,CHKSUM,LOCKTM**  **S LOCKTM=$S(+DILOCKTM>1:DILOCKTM,1:1)**  **L +^TIU(8925,TIUDA,0):LOCKTM**  **E D Q**  **. W:$E(IOST,1,2)="C-" !,"Unable to obtain lock for entry "\_TIUDA**  **S DIE=8925,DA=TIUDA**  **S DR=".01////^S X="\_TIUTO\_";.04////^S X="\_$$DOCCLASS^TIULC1(TIUTO)**  **D ^DIE**  **S CHKSUM=+$$CHKSUM^TIULC("^TIU(8925,"\_+TIUDA\_",""TEXT"")")**  **D AUDIT^TIUEDI1(TIUDA,CHKSUM,CHKSUM)**  **L -^TIU(8925,TIUDA,0)**  **W:$E(IOST,1,2)="C-" !,"Entry demoted from posting status "\_TIUDA**  **Q** |

###### Routine TIUCWDUT

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | TIUCWDUT | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | CWAD/Postings Auto-Demotion Setup [TIU CWAD AUTO-DEMOTION] | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | $$GET1^DIQ, GETLST^XPAR, ^DIC, $$POSTSIGN^TIULC1, ^DIR |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **TIUCWDUT ;SLC/TDP - CWAD POSTINGS AUTO-DEMOTION UTILITY ;10/27/15 15:11**  **;;1.0;TEXT INTEGRATION UTILITIES;\*\*291\*\*;Jun 20, 1997;Build 4**  **Q**  **EN ;**  **W @IOF**  **N TIUCWAD,TIUNON,TIUNOGO,STRING,POSTTL,RESTRICT**  **;I DUZ(0)'["@" D Q**  **;. W !,"Because of TIU file #8925.1 security, programmer access with '@' is required for this option to work."**  **D GETSCRN(.POSTTL,.RESTRICT)**  **S TIUCWAD=$$GETCWAD(.POSTTL) Q:TIUCWAD'>0 W !**  **S TIUNOGO=$$INHERIT(TIUCWAD) I TIUNOGO W !,"Okay. Nothing will be changed." Q**  **S TIUNON=$$GETNON(.POSTTL,.RESTRICT) Q:TIUNON'>0**  **K POSTTL,RESTRICT**  **S STRING="D SILENT^TIUCWD("\_TIUCWAD\_","\_TIUNON\_")"**  **;D CHECK Q:PREVPOST]""**  **D FILE(TIUCWAD,STRING)**  **W !!,"Done. Post-Signature code has been set (or reset) as follows..."**  **W !,"TITLE: "\_$$GET1^DIQ(8925.1,TIUCWAD,.01)**  **W !,"POST-SIGNATURE ACTION: "\_$$GET1^DIQ(8925.1,TIUCWAD,4.9)**  **Q**  **GETSCRN(POSTTL,RESTRICT) ;Set-up arrays used in ^DIC call screen**  **N CWD,CWDEXC,CWDIEN,POSTDC,POSTLETT,QUIT,VHATITLE,X**  **D GETLST^XPAR(.CWD,"PKG","TIU CWAD EXCLUDED TITLES","I")**  **S X=""**  **I $D(CWD) D**  **. F S X=$O(CWD(X)) Q:X="" D**  **.. S CWDIEN=+$G(CWD(X))**  **.. I CWDIEN>0 D**  **... S CWDEXC(CWDIEN)=""**  **. K CWD**  **F POSTLETT="C","W","A","D" S POSTDC=$O(^TIU(8925.1,"APOST",POSTLETT,0)) I +POSTDC S POSTDC(POSTDC)=""**  **S POSTDC=0**  **F S POSTDC=$O(POSTDC(POSTDC)) Q:'+POSTDC D**  **. S POSTTL=0**  **. F S POSTTL=+$O(^TIU(8925.1,POSTDC,10,"B",POSTTL)) Q:'+POSTTL D**  **.. S VHATITLE=+$P($G(^TIU(8925.1,POSTTL,15)),U,1)**  **.. S QUIT=0**  **.. I $D(CWDEXC(VHATITLE)) S RESTRICT(POSTTL)="",QUIT=1**  **.. I QUIT=0 S POSTTL(POSTTL)=""**  **Q**  **GETCWAD(POSTTL) ;Select CWAD type documents**  **N DIC,DLAYGO,DTOUT,DUOUT,X,Y**  **S DIC=8925.1,DIC(0)="AEMQ",DIC("A")="Select a CWAD/Postings TITLE for auto-demotion: "**  **S DIC("S")="I $D(POSTTL(Y))"**  **D ^DIC**  **Q +Y**  **GETNON(POSTTL,RESTRICT) ;Select non-CWAD type documents**  **N DIC,DLAYGO,DTOUT,DUOUT,X,Y**  **S DIC=8925.1,DIC(0)="AEMQ",DIC("A")="Select a NON-Posting TITLE as the demotion target: "**  **S DIC("S")="I ($P($G(^TIU(8925.1,Y,0)),U,4)=""DOC"")&('$D(POSTTL(Y))&('$D(RESTRICT(Y))))"**  **D ^DIC**  **Q +Y**  **CHECK ;This subroutine is being orphaned as of 8-28-09**  **;S PREVPOST=$$POSTSIGN^TIULC1(TIUCWAD)**  **;I PREVPOST]"" D**  **;. W !,"The selected CWAD/Postings title already has a TIU POST-SIGNATURE ACTION,"**  **;. W !,"possibly by inheritance."**  **;. W !!,"It is --> ",PREVPOST**  **;. W !,"You will have to set the code manually through programmer-access FileMan."**  **;Q**  **INHERIT(TIUCWAD) ;Previous Post-Signature Action**  **;Returns a 1 to indicate No-Go, meaning do not proceed**  **N DA,DIR,DIROUT,DIRUT,DTOUT,DUOUT,PREVPOST,X,Y**  **S Y=0**  **S PREVPOST=$$POSTSIGN^TIULC1(TIUCWAD)**  **Q:PREVPOST']"" 0**  **W !,"The selected CWAD/Postings title already has a TIU POST-SIGNATURE ACTION,"**  **W !,"possibly by inheritance."**  **W !!,"It is --> ",PREVPOST**  **W !!,"Do you wish to CONTINUE (and thus OVERWRITE) the Post-Signature Action?"**  **W !,"You can also choose to exit, which will make no changes.",!**  **S DIR(0)="YA",DIR("A")="Continue/overwrite? (Y/N): "**  **D ^DIR**  **Q $S(+Y'=1:1,1:0)**  **FILE(TIUCWAD,STRING) ;Saves new Post-Signature Action**  **;Changing to a direct file save to get past file protections.**  **;N DA,DIDEL,DIE,DR,DTOUT,BKPDUZ0**  **;S DIE=8925.1,DA=TIUCWAD,DR="4.9///^S X=STRING" D ^DIE**  **S ^TIU(8925.1,TIUCWAD,4.9)=STRING**  **Q** |

##### NSR# 20110903 - IV infusion administration not accurately documented in CPRS logs

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWPS | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | Order Details display from Orders Tab | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORDV06B | ^%DT, ORCXPDN1, ORWDXR, ORX8, PSBMLHS, PSBO, PSORRL, VADPT, XLFDT, XPAR, XPDUTL |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | 3448 and 3459 | | | | |
| **Data Passing** | Input | Output Reference | Both | Global Reference | ocal |
| **Input Attribute Name and Definition** | Name:  Definition: | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| Current Logic |
| --- |
| N/A |

| Modified Logic (Changes are in bold) |
| --- |
| **S ^TMP("PSBMH",$J)=""**  I (ORPHMID["P")!(ORPHMID="") D **K ^TMP("PSBMH",$J)** Q  . I '$L($T(HISTORY^PSBMLHS)) D Q  . . S @ORROOT@(0)="This report is only available using BCMA version 2.0  ."  . D HISTORY^PSBMLHS(.ORROOT,DFN,ORPSID) ; DBIA #3459 for BCMA v2.0  ; If the order has a Display Group of IV MEDICATION the use the Pharmac  y order number to get the MA  I ($P($G(^OR(100,+ORIFN,0)),U,11)=ISIV)!($P($G(^OR(100,+ORIFN,0)),U,11)  =HPIV)!($P($G(^OR(100,+ORIFN,0)),U,11)=CLIVDISP) D **K ^TMP("PSBMH",$J)** Q  . I 'CKPKG S @ORROOT@(0)="Medication Administration History is not avai  lable at this time for IV fluids."  . I CKPKG D  . . D RPC^PSBO(.ORROOT,"PM",DFN,"","","","","","","","","",ORPHMID) ;D  BIA #3955  . . I '$D(@ORROOT) S @ORROOT@(0)="No Medication Administration History  found for the IV order."  I '$L($T(HISTORY^PSBMLHS)) D **K ^TMP("PSBMH",$J)** Q  . S @ORROOT@(0)="This report is only available using BCMA version 2.0."  D HISTORY^PSBMLHS(.ORROOT,DFN,ORPSID) ; DBIA #3459 for BCMA v2.0  **K ^TMP("PSBMH",$J)** Q |

###### Routine PSBOPM

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSBOPM | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | Medication History Reports | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Reports | %DT, %DTC, DIC, DIQ, PSBCSUTX, PSBOHDR, PSBOML, PSBOPM1, PSJBCMA1, XPAR |

| Routines | Activities | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | N/A | | | | |
| **Related Protocols** | N/A | | | | |
| **Related Integration Control Registrations (ICRs)** | 3448, 3459 | | | | |
| **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 |
| --- |
| GETADSO ; GET ALL ADDITIVES FOR ALL ORDERABLE ITEMS  K PSBAOUT,PSBSOUT  …..  ; GET ALL SOLUTIONS FOR ALL ORDERABLE ITEMS  S XA="" F S XA=$O(TMP("PSBOIS",$J,XA)) Q:XA="" D  .D LIST^DIC(52.7,"","@;9I","QPI","","","","AOI","","","PSBSOUT")  .S XB=0 F S XB=$O(PSBSOUT("DILIST",XB)) Q:XB="" D  ..I $P(PSBSOUT("DILIST",XB,0),"^",2)=XA D  ...S TMP("PSBSOLS",$J,$P(PSBSOUT("DILIST",XB,0),"^",1))="" |

| Modified Logic (Changes are in bold) |
| --- |
| GETADSO ; GET ALL ADDITIVES FOR ALL ORDERABLE ITEMS  K PSBAOUT,PSBSOUT**,PSBSOL**  … .  ; GET ALL SOLUTIONS FOR ALL ORDERABLE ITEMS  S XA="" F S XA=$O(TMP("PSBOIS",$J,XA)) Q:XA="" D  .D LIST^DIC(52.7,"","@;9I","QPI","","","","AOI","","","PSBSOUT")  .S XB=0 F S XB=$O(PSBSOUT("DILIST",XB)) Q:XB="" D  ..I $P(PSBSOUT("DILIST",XB,0),"^",2)=XA D  ...**S PSBSOL=$P(PSBSOUT("DILIST",XB,0),"^") I $D(^TMP("PSBMH",$J)),$P(^P**  **S(52.7,PSBSOL,0),"^",14)'=1 Q**  **...S TMP("PSBSOLS",$J,PSBSOL)=""**  K PSBSOUT  Q |

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSB82PST | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | Post install for PSB\*3\*82. New version of ORDERCOM.DLL | | | |

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

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

| Current Logic |
| --- |
| N/A |

| Modified Logic (Changes are in bold) |
| --- |
| **PSB82PST;;SLC/ARF - Post install for ORDERCOM.DLL ;10/26/17 14:26**  **;;3.0;BAR CODE MED ADMIN;\*\*82\*\*;Mar 2004;**  **Q**  **; This routine uses the following IAs:**  **; 2263 - ^XPAR (supported)**  **;**  **EN ; main entry point**  **D DLL**  **Q**  **DLL ;**  **; Variables:**  **; MOBDLL [Private] Current version of DLL being installed**  **;**  **; New private variables**  **N MOBDLL**  **; Announce my intentions**  **D BMES^XPDUTL("Updating DLL parameter.")**  **S MOBDLL="2.0.17.1" ;patch 82**  **D EN^XPAR("SYS","PSB MOB DLL REQUIRED VERSION",MOBDLL,1)**  **Q** |

##### NSR# 20071211 Changes to Allergy/Pharmacy Packages

**6.2.2.2.13.1. Routine ORQ2**

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORQ2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | DETAIL^ORWOR | NMSP^ORCD  TEXT^ORQ12  GET^XPAR  RAD^ORQ21  GETDLG^ORCD  REPEAT^XLFSTR  BA^ORQ21  USER^ORQ20  DIWP  DATE^ORQ20 |

| 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 |
| --- |
| D5 K ^TMP($J,"OCDATA") I $$OCAPI^ORCHECK(+ORIFN,"OCDATA") D  . N CK,OK,X0,X,CDL,I S CNT=CNT+1,@ORY@(CNT)="Order Checks:"  . D:$D(IOUON) SETVIDEO(CNT,1,13,IOUON,IOUOFF)  . S CK=0 F S CK=$O(^TMP($J,"OCDATA",CK)) Q:CK'>0 D  .. S X0=^TMP($J,"OCDATA",CK,"OC NUMBER")\_U\_^TMP($J,"OCDATA",CK,"OC LEVEL")\_U\_U\_^TMP($J,"OCDATA",CK,"OR REASON")\_U\_^TMP($J,"OCDATA",CK,"OR PROVIDER")\_U\_^TMP($J,"OCDATA",CK,"OR DT")  .. S X=^TMP($J,"OCDATA",CK,"OC TEXT",1,0)  .. S CDL=$$CDL($P(X0,U,2)) I $P(X0,U,6),'$D(OK) S OK=$P(X0,U,4,6)  .. I $L(X)'>68 S CNT=CNT+1,@ORY@(CNT)=CDL\_X D XTRA Q  .. S DIWL=1,DIWR=68,DIWF="C68" K ^UTILITY($J,"W") D ^DIWP  .. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=CDL\_^(I,0),CDL=" "  .. D XTRA  . K ^TMP($J,"OCDATA")  . Q:'$L($G(OK)) S CNT=CNT+1,@ORY@(CNT)="Override: "\_$S($P(OK,U,2):$$USER^ORQ20($P(OK,U,2))\_" on",1:"")\_$$DATE^ORQ20($P(OK,U,3))  . I $L($P(OK,U))'>68 S CNT=CNT+1,@ORY@(CNT)=" "\_$P(OK,U) Q  . S DIWL=1,DIWR=68,DIWF="C68",X=$P(OK,U) K ^UTILITY($J,"W") D ^DIWP  . S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=" "\_^(I,0)  K ^TMP("ORWORD",$J),^UTILITY($J,"W")  Q |

| Modified Logic (Changes are in bold) |
| --- |
| D5 K ^TMP($J,"OCDATA") I $$OCAPI^ORCHECK(+ORIFN,"OCDATA") D  . N CK,OK,X0,X,CDL,I,**ACK,ALLERGY,HDR S HDR=0**  . D:$D(IOUON) SETVIDEO(CNT,1,13,IOUON,IOUOFF)  . **D ALLERGY ;405**  . S CK=0 F S CK=$O(^TMP($J,"OCDATA",CK)) Q:CK'>0 D  .. **Q:$D(ALLERGY(CK))**  .. **S:HDR=0 CNT=CNT+1,@ORY@(CNT)=" ",CNT=CNT+1,@ORY@(CNT)="Order Checks:",HDR=1**  .. S X0=^TMP($J,"OCDATA",CK,"OC NUMBER")\_U\_^TMP($J,"OCDATA",CK,"OC LEVEL")\_U\_U\_^TMP($J,"OCDATA",CK,"OR REASON")\_U\_^TMP($J,"OCDATA",CK,"OR PROVIDER")\_U\_^TMP($J,"OCDATA",CK,"OR DT")  .. S X=^TMP($J,"OCDATA",CK,"OC TEXT",1,0)  .. S CDL=$$CDL($P(X0,U,2)) I $P(X0,U,6),'$D(OK) S OK=$P(X0,U,4,6)  .. I $L(X)'>68 S CNT=CNT+1,@ORY@(CNT)=CDL\_X D XTRA Q  .. S DIWL=1,DIWR=68,DIWF="C68" K ^UTILITY($J,"W") D ^DIWP  .. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=CDL\_^(I,0),CDL=" "  .. D XTRA  . K ^TMP($J,"OCDATA")  . Q:'$L($G(OK)) S CNT=CNT+1,@ORY@(CNT)="Override: "\_$S($P(OK,U,2):$$USER^ORQ20($P(OK,U,2))\_" on ",1:"")\_$$DATE^ORQ20($P(OK,U,3))  . I $L($P(OK,U))'>68 S CNT=CNT+1,@ORY@(CNT)=" "\_$P(OK,U) Q  . S DIWL=1,DIWR=68,DIWF="C68",X=$P(OK,U) K ^UTILITY($J,"W") D ^DIWP  . S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=" "\_^(I,0)  K ^TMP("ORWORD",$J),^UTILITY($J,"W"),ALLERGY  Q  **ALLERGY ;separate the ALLERGY-DRUG INTERACTION Order Checks (405)**  **N ORRET,ORINSTAN,ORSAVINS ;NEW**  **S ORRET=1 ;NEW**  **S (ACK,CK)=0 F S CK=$O(^TMP($J,"OCDATA",CK)) Q:CK'>0 D**  **. I $G(^TMP($J,"OCDATA",CK,"OC NUMBER"))'=3 Q**  **. S ALLGYDRG(CK)=" "**  **. I ACK=0 S CNT=CNT+1,@ORY@(CNT)=" ",CNT=CNT+1,@ORY@(CNT)="Allergy Order Checks:",CNT=CNT+1,ACK=1**  **. S X0=^TMP($J,"OCDATA",CK,"OC NUMBER")\_U\_^TMP($J,"OCDATA",CK,"OC LEVEL")\_U\_U\_^TMP($J,"OCDATA",CK,"OR REASON")\_U\_^TMP($J,"OCDATA",CK,"OR PROVIDER")\_U\_^TMP($J,"OCDATA",CK,"OR DT")**  **. S X=^TMP($J,"OCDATA",CK,"OC TEXT",1,0)**  **. S CDL=$$CDL($P(X0,U,2)) I $P(X0,U,6),'$D(OK) S OK=$P(X0,U,4,6)**  **. I $L(X)'>68 S CNT=CNT+1,@ORY@(CNT)=CDL\_X D XTRA Q**  **. S DIWL=1,DIWR=68,DIWF="C68" K ^UTILITY($J,"W") D ^DIWP**  **. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=CDL\_^(I,0),CDL=" "**  **. S ORINSTAN=$G(^TMP($J,"OCDATA",CK,"OC INSTANCE")) ;NEW**  **. I ORINSTAN>0 D**  **.. I $$GET1^DIQ(100.517,ORRET\_","\_ORINSTAN\_",",11)'="" D ;NEW**  **... S:'$D(ORSAVINS) ORSAVINS=ORINSTAN**  **. D XTRA**  **I ACK=1 D**  **. Q:'$L($G(OK)) S CNT=CNT+1,@ORY@(CNT)="Override: "\_$S($P(OK,U,2):$$USER^ORQ20($P(OK,U,2))\_" on ",1:"")\_$$DATE^ORQ20($P(OK,U,3))**  **. I $L($P(OK,U))'>68 S CNT=CNT+1,@ORY@(CNT)=" "\_$P(OK,U),CNT=CNT+1 Q**  **. S DIWL=1,DIWR=68,DIWF="C68",X=$P(OK,U) K ^UTILITY($J,"W") D ^DIWP**  **. S I=0 F S I=$O(^UTILITY($J,"W",DIWL,I)) Q:I'>0 S CNT=CNT+1,@ORY@(CNT)=" "\_^(I,0)**  **I $D(ORSAVINS) D**  **. S CNT=CNT+1,@ORY@(CNT)=" "**  **. S CNT=CNT+1,@ORY@(CNT)="Remote Comment: "\_$$GET1^DIQ(100.517,ORRET\_","\_ORSAVINS\_",",11)**  **Q** |

| **Routines** | **Activities** | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSSPOIMO | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | Edit Orderable Items [PSS EDIT ORDERABLE ITEMS] | | | |

| **Related Routines** | **Routines “Called By”** | **Routines “Called”** |
| --- | --- | --- |
|  |  | DD^%DT, EN^DDIOL, ^DIC, MIX^DIC1, ^DIE,UPDATE^DIE, ^DIK, $$GET1^DIQ, ^DIR, EN2^PSSHL1,MEDRT^PSSJORDF, CHECK^PSSPOID2,EN^PSSPOIDT, REST^PSSPOIDT, ADDSYN^PSSPOIMP |

| **Routines** | **Activities** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^PS(50.7) | | | | |
| **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** |
| --- |
| I $G(PSINORDE)="I" I $O(PSSDACT(0))!($O(PSSSACT(0)))!($O(PSSAACT(0))) D  REST^PSSPOIDT(PSOIEN)  S DIK="^PS(50.7,",DA=PSOIEN,DIK(1)=.04 D EN^DIK K DIK  K PSBEFORE,PSAFTER,PSINORDE,PSSDTENT,PSSDACT,PSSDACTI,PSSSACT,PSSSACTI,  PSSAACT,PSSAACTI |

| **Modified Logic (Changes are in bold)** |
| --- |
| I $G(PSINORDE)="I" I $O(PSSDACT(0))!($O(PSSSACT(0)))!($O(PSSAACT(0))) D REST^PSSPOIDT(PSOIEN)  S DIK="^PS(50.7,",DA=PSOIEN,DIK(1)=.04 D EN^DIK K DIK  K PSBEFORE,PSAFTER,PSINORDE,PSSDTENT,PSSDACT,PSSDACTI,PSSSACT,PSSSACTI,  PSSAACT,PSSAACTI  **N DIE,DA,DR ; input Indications for Use**  **S DIE="^PS(50.7,",DA=PSOIEN,DR="14;13" D ^DIE K DIE** |

| **Routines** | **Activities** | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSSPOIM1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** | Dispense Drug/Orderable Item Maintenance [PSS MAINTAIN ORDERABLE ITEMS]  Drug Enter/Edit [PSS DRUG ENTER/EDIT] | | | |

| **Related Routines** | **Routines “Called By”** | **Routines “Called”** |
| --- | --- | --- |
|  |  | MRSEL^PSSPOIMO, REST^PSSPOIDT, ^DIE |

| **Routines** | **Activities** | | | | |
| --- | --- | --- | --- | --- | --- |
| **Data Dictionary (DD) References** | ^PS(50.7) | | | | |
| **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** |
| --- |
| D ^DIE S PSAFTER=$P(^PS(50.7,PSVAR,0),"^",4) K DIE,DA,DR,PSSOTH,^TMP("PSJMR",$J),^TMP("PSSDMR",$J) I $D(PSSOU),'$G(PSSOU) D MRSEL^PSSPOIMO K ^TMP("PSJMR",$J)  S:PSBEFORE&('PSAFTER) PSINORDE="D" S:PSAFTER PSINORDE="I"  I PSINORDE'="" D REST^PSSPOIDT(PSVAR)  K PSBEFORE,PSBEFORE1,PSAFTER,PSINORDE |

| **Modified Logic (Changes are in bold)** |
| --- |
| D ^DIE S PSAFTER=$P(^PS(50.7,PSVAR,0),"^",4) K DIE,DA,DR,PSSOTH,^TMP("PSJMR",$J),^TMP("PSSDMR",$J) I $D(PSSOU),'$G(PSSOU) D MRSEL^PSSPOIMO K ^TMP("PSJMR",$J)  S:PSBEFORE&('PSAFTER) PSINORDE="D" S:PSAFTER PSINORDE="I"  I PSINORDE'="" D REST^PSSPOIDT(PSVAR)  K PSBEFORE,PSBEFORE1,PSAFTER,PSINORDE  **N DIE,DA,DR ; Indications for Use fields PSS\*1\*204**  **S DIE="^PS(50.7,",DA=PSVAR,DR="14;13" D ^DIE K DIE** |

Routine PSS50P7

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | PSS50P7 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | ORWDPS2 |  |

| 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 |
| --- |
| SETSCRN ;Set Screen for inactive entries in PHARMACY ORDERABLE ITEM file (#50.7).  ;Naked reference below refers to ^PS(50.7,+Y,0)  S SCR("S")="S ND=$P($G(^(0)),U,4) I ND=""""!(ND>PSSFL)"  Q |

| Modified Logic (Changes are in bold) |
| --- |
| SETSCRN ;Set Screen for inactive entries in PHARMACY ORDERABLE ITEM file (#50.7).  ;Naked reference below refers to ^PS(50.7,+Y,0)  S SCR("S")="S ND=$P($G(^(0)),U,4) I ND=""""!(ND>PSSFL)"  Q  **INDCATN(PSSIEN,LIST) ; return Indications for Use of Prescription and Medication Orders**  **N DEF,I,IND,K**  **K ^TMP($J,LIST) S DEF=""**  **I $P($G(^PS(50.7,PSSIEN,4)),"^",2)'="" S DEF=1,IND=$P(^PS(50.7,PSSIEN,4),"^",2),^TMP($J,LIST,1)=IND\_"^"\_DEF ;if most common indication exists, return its value**  **S I=0 F S I=$O(^PS(50.7,PSSIEN,"IND",I)) Q:'I D**  **.S IND=$G(^PS(50.7,PSSIEN,"IND",I,0)) ;if indication exists, return it in an array**  **.S K=$S(DEF=1:I+1,1:I)**  **.S ^TMP($J,LIST,K)=IND**  **Q** |

Routine ORWDPS2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORWDPS2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  |  | PSS50P7 |

| 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 |
| --- |
| . ; start, expires, next admin  I PSTYPE="O" D  . ; days supply, quantity, refills |

| Modified Logic (Changes are in bold) |
| --- |
| . ; start, expires, next admin  I PSTYPE="O" D  . ; days supply, quantity, refills  **S ILST=ILST+1,LST(ILST)="~Indication" D INDICAT**  **INDICAT ; from OISLCT return Indication for Use of Prescription or Medication Order**  **N G,IND**  **D INDCATN^PSS50P7(OI,"ORWDPIND")**  **S G="" F S G=$O(^TMP($J,"ORWDPIND",G)) Q:G="" D**  **.S INDCAT=$G(^TMP($J,"ORWDPIND",G))**  **.S IND=$S($P(INDCAT,"^",2)=1:"d"\_$P(INDCAT,"^"),1:"i"\_INDCAT)**  **.S ILST=ILST+1,LST(ILST)=IND**  **Q** |

##### NSR # 200120404 – Adverse Reaction Reporting File Modifications

###### Routine GMRADPT

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified PASS tag:**  PASS(GMRAREC,GMRAL) ;RETRIEVE LOCAL DATA  ;PARAMETERS: GMRAREC => IEN OF THE CURRENT ALLERGY  ; GMRAL => ARRAY IN WHICH TO RETURN DATA  N GMRANODE,%,GMRAX,GMRAY,GMRAZ,GMRAKC  I '$D(MECH) D  .D MECH  .S GMRAKC=1  S GMRANODE=$G(^GMR(120.8,GMRAREC,0)) Q:GMRANODE=""  S %=$P(GMRANODE,U,14)  S GMRAL(GMRAREC)=$P(GMRANODE,U,1,2)\_U\_$E($P(GMRANODE,U,20))\_U\_+$P(GMRAN  ODE,U,16)\_U\_$S(%="A"!(%="U"):0,1:1)  S GMRAL(GMRAREC)=GMRAL(GMRAREC)\_U\_$S(%="A":"ALLERGY;0",%="P":"PHARMACOL  OGIC;2",%="U":"UNKNOWN;U",1:"")\_U\_$P(GMRANODE,U,20)\_U\_$S(%'="":$G(MECH(%)),1:"")  S GMRAL(GMRAREC)=GMRAL(GMRAREC)\_U\_$P(GMRANODE,U,3)  ;\*BD  I '$G(GMRAV1) D  .;\*ED  .S %=$P(GMRANODE,U,6)  .S GMRAL(GMRAREC)=GMRAL(GMRAREC)\_U\_$$EXTERNAL^DILFD(120.8,6,,%)\_";"\_%  .I $D(^GMR(120.85,"C",GMRAREC))>9 D  ..N IEN,IDX  ..S IEN=0 F S IEN=$O(^GMR(120.85,"C",GMRAREC,IEN)) Q:'+IEN D  ...S IDX=1+$G(IDX),%=$P($G(^GMR(120.85,IEN,0)),U,14)  ...S GMRAL(GMRAREC,"O",IDX)=$$EXTERNAL^DILFD(120.85,14.5,,%)\_";"\_%\_U  ...S %=$P($G(^GMR(120.85,IEN,0)),U)  ...S GMRAL(GMRAREC,"O",IDX)=GMRAL(GMRAREC,"O",IDX)\_$$EXTERNAL^DILFD(120  .85,.01,,%)\_";"\_%  **I $P(GMRANODE,U,6)="h" D**  **.N SEVR,SEVRDT**  **.S SEVR=$P(GMRANODE,U,8)**  **.S SEVRDT=$P(GMRANODE,U,9)**  **.S GMRAL(GMRAREC,"H")=$$EXTERNAL^DILFD(120.8,8,,SEVR)\_";"\_SEVR\_U\_$$EXTE**  **RNAL^DILFD(120.8,9,,SEVRDT)\_";"\_SEVRDT**  I $O(^GMR(120.8,GMRAREC,10,0)) D  .S GMRAX=0,GMRAY=1 F S GMRAX=$O(^GMR(120.8,GMRAREC,10,GMRAX)) Q:GMRAX<  1 D  ..S GMRAZ=$G(^GMR(120.8,GMRAREC,10,GMRAX,0))  ..Q:GMRAZ=""  ..S GMRAZ(1)=$S(+GMRAZ'=GMRAOTH:$P($G(^GMRD(120.83,+GMRAZ,0)),U)\_";"\_+G  MRAZ,1:$P(GMRAZ,U,2)\_";"\_+GMRAZ)  ..;\*BD  ..I '$G(GMRAV1) D  ...;\*ED (CLEAN UP PERIODS)  ...S GMRAZ(1)=GMRAZ(1)\_U\_$$FMTE^XLFDT($P(GMRAZ,U,4))\_";"\_$P(GMRAZ,U,4)  ..S GMRAL(GMRAREC,"S",GMRAY)=GMRAZ(1),GMRAY=GMRAY+1  K:+$G(GMRAKC) MECH  Q |

###### Routine GMRAGUI

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified OBHIST tag:**  OBSHIST S STRING="~OBS/HIST" D NEXT  S GMRAOBS=$P(GMRA(0),U,6)  S STRING="d"\_GMRAOBS\_"^"\_$S(GMRAOBS="o":"OBSERVED",GMRAOBS="h":"HISTORI  CAL",1:"") D NEXT  **I GMRAOBS="o" D EN1^GMRAGUI1**  **I GMRAOBS="h",(($P(GMRA(0),U,8)'="")!($P(GMRA(0),U,9)'="")) D**  **.S STRING="~HISTORICAL" D NEXT**  **.S Y=$P(GMRA(0),U,9) X ^DD("DD")**  **.S STRING="tDate/Time of Event: "\_Y D NEXT**  **.S SEVCOD=$P(GMRA(0),U,8)**  **.S SEVER=$S(SEVCOD=1:"MILD",SEVCOD=2:"MODERATE",SEVCOD=3:"SEVERE",1:"")**  **.S STRING="tSeverity : "\_SEVER D NEXT** |

###### Routine GMRAGUI1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAGUI1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified UPDATE tag:**  UPDATE(GMRAIEN,DFN,GMRARRAY) ;Add/edit allergies  N NEW,NKA,FDA,NODE,IEN,SUB,FILE,DA,DIK,SIEN,GMRAS0,GMRAL,GMRAPA,GMRAAR,  GMRALL,GMRADFN,GMRAOUT,GMRAROT,GMRAPN  S NEW='$G(GMRAIEN)  I NEW,$$DUPCHK^GMRAOR0(DFN,$P(@GMRARRAY@("GMRAGNT"),U))=1 S ORY="-1^Pat  ient already has a "\_$P(@GMRARRAY@("GMRAGNT"),U)\_" reaction entered. No duplica  tes allowed." Q  L +^XTMP("GMRAED",DFN):1 I '$T D MESS Q  D SITE^GMRAUTL S GMRASITE(0)=$G(^GMRD(120.84,+GMRASITE,0))  S NKA='$$NKA^GMRANKA(DFN) ;is patient NKA?  I NKA,NEW D  .S FDA(120.86,"?+"\_DFN\_",",.01)=DFN  .S FDA(120.86,"?+"\_DFN\_",",1)=1  .S FDA(120.86,"?+"\_DFN\_",",2)=DUZ  .S FDA(120.86,"?+"\_DFN\_",",3)=$G(@GMRARRAY@("GMRAORDT"),$$NOW^XLFDT)  .S IEN(DFN)=DFN  .D UPDATE^DIE("","FDA","IEN")  K FDA,IEN  S NODE=$S($G(NEW):"+1,",1:(GMRAIEN\_","))  S:$G(NEW) FDA(120.8,NODE,.01)=DFN  I $P($G(@GMRARRAY@("GMRAGNT")),U,2)["50.67" S $P(@GMRARRAY@("GMRAGNT"),  U,2)=$$TGTOG^PSNAPIS($P(@GMRARRAY@("GMRAGNT"),U))\_";PSNDF(50.6,"  F SUB="GMRAGNT;.02","GMRATYPE;3.1","GMRANATR;17","GMRAORIG;5","GMRAORDT  ;4","GMRAOBHX;6" D  .S FDA(120.8,NODE,$P(SUB,";",2))=$P(@GMRARRAY@($P(SUB,";")),U)  .I (SUB["GMRAGNT"),NEW S FDA(120.8,NODE,1)=$P(@GMRARRAY@($P(SUB,";")),U  ,2)  **.I SUB["GMRAOBHX",$P(@GMRARRAY@("GMRAOBHX"),U)="h" D**  **..S FDA(120.8,NODE,8)=$G(@GMRARRAY@("GMRASEVR"))**  **..S FDA(120.8,NODE,9)=$G(@GMRARRAY@("GMRARDT"))**  D UPDATE^DIE("","FDA","IEN")  S:NEW GMRAIEN=IEN(1)  K FDA  F SUB="GMRACHT","GMRAIDBN" D  .Q:'$D(@GMRARRAY@(SUB)) ;Stop if no updates  .S FILE=$S(SUB="GMRACHT":120.813,1:120.814)  .S FDA(FILE,"+1,"\_GMRAIEN\_",",.01)=@GMRARRAY@(SUB,1)  .S FDA(FILE,"+1,"\_GMRAIEN\_",",1)=DUZ  .D UPDATE^DIE("","FDA")  I $D(@GMRARRAY@("GMRACMTS")) D ADCOM(GMRAIEN,"O",$NA(@GMRARRAY@("GMRACM  TS"))) ;Add comments if included  K FDA  S SUB=0 F S SUB=$O(@GMRARRAY@("GMRASYMP",SUB)) Q:'+SUB D  .S GMRAS0=^(SUB) ;Naked from above  .Q:$P(^(SUB),U)="" ;25 No text or free text entered so don't store  .S SIEN=$O(^GMR(120.8,GMRAIEN,10,"B",$P(GMRAS0,U),0))  .I SIEN,$P(^GMR(120.8,GMRAIEN,10,SIEN,0),U,4)=$P(GMRAS0,U,3) Q ;Exists  and nothing has changed  .I SIEN,$P(GMRAS0,U,5)="@" S DIK="^GMR(120.8,"\_GMRAIEN\_",",DA(1)=GMRAIE  N,DA=SIEN D ^DIK Q ;Sign/symptom deleted  .S:'SIEN FDA(120.81,"+1,"\_GMRAIEN\_",",.01)=$S($P(GMRAS0,U)="FT":$O(^GMR  D(120.83,"B","OTHER REACTION",0)),1:$P(GMRAS0,U))  .S NODE=$S(SIEN:SIEN\_","\_GMRAIEN,1:"+1,"\_GMRAIEN\_",")  .S:$P(GMRAS0,U)="FT" FDA(120.81,NODE,1)=$P(GMRAS0,U,2)  .S FDA(120.81,NODE,2)=DUZ  .S FDA(120.81,NODE,3)=$P(GMRAS0,U,3)  .D UPDATE^DIE("","FDA","","ERR")  .S GMRAROT($P(GMRAS0,U,2))="" ;21 record s/s added  I NEW D  .S GMRALL(GMRAIEN)="" D VAD^GMRAUTL1(DFN,,.GMRALOC,.GMRANAM) D EN7^GMRA  MCB ;Send mark chart/ID band bulletin if needed.  .I $P(@GMRARRAY@("GMRAOBHX"),U)="o" D ;if observed reaction add data t  o 120.85  ..S GMRAOUT=0 ;21  ..S GMRAL(GMRAIEN,"O",GMRAIEN)=$G(@GMRARRAY@("GMRARDT"))\_"^"\_$G(@GMRARR  AY@("GMRASEVR"))  ..S GMRADFN=DFN |
| ..S GMRAL(GMRAIEN)="^^"\_$P($G(@GMRARRAY@("GMRAGNT")),U)\_"^^^^"\_$G(@GMRA  RRAY@("GMRAORIG"))  ..M GMRAL(GMRAIEN,"S")=@GMRARRAY@("GMRASYMP")  ..S SUB=0 F S SUB=$O(GMRAL(GMRAIEN,"S",SUB)) Q:'+SUB S $P(GMRAL(GMRAI  EN,"S",SUB),U,2)=$P(GMRAL(GMRAIEN,"S",SUB),U,2)\_"^" S:$P(GMRAL(GMRAIEN,"S",SUB),  U)="FT" $P(GMRAL(GMRAIEN,"S",SUB),U)=$O(^GMRD(120.83,"B","OTHER REACTION",0))  ..S GMRAL=GMRAIEN  ..D ADVERSE^GMRAOR7(GMRAIEN,.GMRAL) ;adds entry to 120.85  ..S GMRAIEN(GMRAIEN)="" ;21  ..D EN1^GMRAPET0(GMRADFN,.GMRAIEN,"S",.GMRAOUT) ;21 File progress note  ..I $G(@GMRARRAY@("GMRATYPE"))["D" S GMRAPA=GMRAIEN D EN1^GMRAPTB ;21 S  end med-watch update  .S GMRAAR=$P($G(@GMRARRAY@("GMRAGNT")),U,2),GMRAPA=GMRAIEN  .D EN1^GMRAOR9 S ^TMP($J,"GMRASF",1,GMRAPA)="" D RANGE^GMRASIGN(1) ;add  ingredients/classes send appropriate bulletins  S ORY=0\_$S(+$G(GMRAPN)>0:("^"\_+$G(GMRAPN)),1:"") ;38 If note was create  d send back IEN  I NEW D GMRACHK^GMRAPEM0(GMRAPA)  L -^XTMP("GMRAED",DFN)  Q |

###### Routine GMRAOR

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified DRUG tag:**  DRUG(DFN,PTR) ;DRUG CHECK  ;PARAMTERS: DFN => IEN IN THE PATIENT FILE (#2)  ; PTR => PSNDA.PSNVPN.LPTR, WHERE  ; PSNDA IS IEN IN VA GENERIC FILE (#50.6),  ; PSNVPN IS IEN IN VA PRODUCT FILE (#50.68), AND  ; LPTR IS IEN IN DRUG FILE (#50)  ;RETURN: 1 => REACTION TO DRUG  ; 0 => NO REACTION TO DRUG  N %,FLG,GMRAC,GMRADR,GMRAI,PSNVPN,PSNDA,LPTR,GMRANVPN,GMRANDA,REACT,CLA  SS,GMRALIST  N TMPFLG,REACS,NODE  S (FLG,TMPFLG)=0,GMRANDA=$P(PTR,"."),GMRANVPN=$P(PTR,".",2),LPTR=$P(PTR  ,".",3)  I GMRANDA>0,($G(@($$NDFREF\_GMRANDA\_",0)"))'="") D  .;INGREDIENT  .S PSNDA=GMRANDA  .I $T(DISPDRG^PSNNGR)]"",GMRANVPN]"" D  ..S NODE="PSNDD",PSNVPN=GMRANVPN  ..K ^TMP(NODE,$J)  ..D DISPDRG^PSNNGR  .I $T(DISPDRG^PSNNGR)=""!(GMRANVPN="") D  ..S NODE="PSN"  ..K ^TMP(NODE,$J)  ..D ^PSNNGR  .S GMRAI=0,%=1 F S GMRAI=$O(^TMP(NODE,$J,GMRAI)) Q:GMRAI<1 I $D(^TMP(  "GMRAOC",$J,"API",GMRAI)) D  ..S FLG=1  ..;\*BD (C U % VAR)  ..I $G(GMRAV1) S GMRAING(%)=^TMP(NODE,$J,GMRAI)\_$$FAC(^TMP("GMRAOC",$J,  "API",GMRAI)),%=%+1  ..I '$G(GMRAV1) D  ...;\*ED (C U DOTS)  ...S REACT="" F S REACT=$O(^TMP("GMRAOC",$J,"API",GMRAI,REACT)) Q:REAC  T="" D  ....S REACS(REACT)=$G(^TMP("GMRAOC",$J,"API",GMRAI,REACT))  ....I $G(REACS(REACT,"ING"))'[GMRAI S REACS(REACT,"ING")=$S($D(REACS(RE  ACT,"ING")):REACS(REACT,"ING")\_"~",1:"")\_^TMP(NODE,$J,GMRAI)\_U\_GMRAI  .K ^TMP(NODE,$J)  .;\*BD  .I $G(GMRAV1),FLG Q  .;\*ED  .;DRUG CLASS  .I GMRANVPN D Q  ..S CLASS=$$DCLCODE^PSNAPIS(GMRANDA,GMRANVPN)  ..S TMPFLG=$$PCCHK(CLASS)  ..S:TMPFLG>0 FLG=TMPFLG  .S GMRALIST=$$CLIST^PSNAPIS(GMRANDA,.GMRALIST) Q:'$G(GMRALIST)  .S GMRALIST=0 F S GMRALIST=$O(GMRALIST(GMRALIST)) Q:'GMRALIST D  ..S TMPFLG=$$PCCHK($P(GMRALIST(GMRALIST),U,2))  ..S:TMPFLG>0 FLG=TMPFLG  ;\*BD  Q:$G(GMRAV1) FLG  ;\*ED  I +LPTR>0 D  .;GMR ALLERGY  .N IEN,NAME  .S IEN=0 F S IEN=$O(^GMR(120.8,"B",DFN,IEN)) Q:'+IEN!(+FLG=3) D  ..I $P($G(^GMR(120.8,IEN,0)),U,3)=(LPTR\_";PSDRUG("),('$P($G(^GMR(120.8,  IEN,"ER")),U)) D  ...D DATA^PSS50(LPTR,,,,,"GMRALST")  ...S NAME=$G(^TMP($J,"GMRALST",LPTR,.01))  ...K ^TMP($J,"GMRALST")  ...D ADDLDATA($NA(REACS(IEN)),IEN)  ...S FLG=3  ...I $G(REACS(IEN,"REC"))'[NAME S REACS(IEN,"REC")=$S($D(REACS(IEN,"REC  ")):"~",1:"")\_NAME\_U\_LPTR  .;DRUG CLASS  .I 'GMRANDA D  ..S TMPFLG=$$PCCHK($$DRP2VACL^GMRAPENC(LPTR))  ..S:TMPFLG>0 FLG=TMPFLG  S:FLG>0 FLG=1  K @RETURN  N STYPE,IPIECE,OPIECE,FIELDS,FIELDS1,ITEM,FILE,NODE,FIELD,MSGNUM,RINDEX  ,SITE,SEV  S STYPE("L")="LOCAL",STYPE("R")="REMOTE"  S FIELDS(1,3)="120.8^4^3",FIELDS(1.1,1)="120.85^.01|14.5+^4",FIELDS(2,1  )="120.81^.01\*^5"  S FIELDS(2,2)="120.8^^6",FIELDS(2,3)="120.8^^7"  S FIELDS(2,3,"OTRANSFORM")="S Y=$S(Y["";PS(50.605,"":$P($G(^PS(50.605,+  Y,0)),U,2),1:$P($G(@(U\_$P(Y,"";"",2)\_+Y\_"",0)"")),U)) S:Y="""" Y=""UNKNOWN"""  S FIELDS(1,4)="120.8^6^8"  S FIELDS1(1.1,1)=""  I $D(REACS)>9 D  .S REACT="" F S REACT=$O(REACS(REACT)) Q:REACT="" D  ..I $P(REACS(REACT),U,8)="h" S FIELDS1(1.1,1)=$G(FIELDS(1.1,1)),FIELDS(  1.1,1)="120.8^9|8+^4"  ..S REACT(1)=$S($P(REACS(REACT),U,7)'="":$P(REACS(REACT),U,7),1:$P(REAC  S(REACT),U,6))  ..S REACT(2)=$P(REACS(REACT),U,8) S:REACT(2)="" REACT(2)="UNKNOWN"  ..S REACT(3)=$P($P(REACS(REACT),U,4),"|",2) S:REACT(3)="" REACT(3)="UNK  NOWN"  ..I $D(RINDEX(REACT(1),REACT(2),REACT(3))) S MSGNUM=$O(RINDEX(REACT(1),  REACT(2),REACT(3),0))  ..E S MSGNUM(0)=1+$G(MSGNUM(0)),MSGNUM=MSGNUM(0),RINDEX(REACT(1),REACT  (2),REACT(3),MSGNUM)=""  ..S @RETURN@(MSGNUM,REACT)=REACS(REACT),SITE=$P(REACS(REACT),U),@RETURN  @(MSGNUM,"MESSAGE",1)=1+$G(@RETURN@(MSGNUM,"MESSAGE",1))  ..F NODE="ING","CLS","REC" I $D(REACS(REACT,NODE)) F ITEM=1:1:$L(REACS(  REACT,NODE),"~") D  ...S $P(@RETURN@(MSGNUM,"MESSAGE","OFFENDERS",NODE),"~",ITEM)=$P($P(REA  CS(REACT,NODE),"~",ITEM),U)  ...S $P(@RETURN@(MSGNUM,REACT,NODE),"~",ITEM)=$P($P(REACS(REACT,NODE),"  ~",ITEM),U,2)  ..S @RETURN@(MSGNUM,"MESSAGE",1,SITE)=$$GET1^DIQ(4,$P(REACS(REACT),U)\_"  ,",.01)\_U\_STYPE($P(REACS(REACT),U,2))\_U  ..S NODE=0 F S NODE=$O(FIELDS(NODE)) Q:NODE="" S OPIECE=0 F S OPIECE  =$O(FIELDS(NODE,OPIECE)) Q:OPIECE="" D  ...S FILE=$P(FIELDS(NODE,OPIECE),U),FIELD=$P(FIELDS(NODE,OPIECE),U,2),I  PIECE=$P(FIELDS(NODE,OPIECE),U,3)  ...I NODE=1 S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",NODE,SITE))  ...I $L(NODE,".")=2 S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",$P(NODE,".")  ,SITE,$P(NODE,".",2)))  ...I NODE'=1,NODE'["." S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",NODE))  ...I FIELD>0,FIELD'["\*",FIELD'["+" S $P(@RETURN,U,OPIECE)=$$EXTERNAL^DI  LFD(FILE,FIELD,,$P(REACS(REACT),U,IPIECE))  ...I 'FIELD,'$D(FIELDS(NODE,OPIECE,"OTRANSFORM")) S $P(@RETURN,U,OPIECE  )=$P(REACS(REACT),U,IPIECE)  ...I $D(FIELDS(NODE,OPIECE,"OTRANSFORM")) N Y S Y=$P(REACS(REACT),U,IPI  ECE) X:Y'="" FIELDS(NODE,OPIECE,"OTRANSFORM") S $P(@RETURN,U,OPIECE)=Y  ...I FIELD["\*" D  ....N TEXT,DELIMIT,ITEMS,STR,COUNT,NEWIVAL  ....S DELIMIT=", ",COUNT=0,COUNT(1)=1  ....I $P($G(@RETURN),U,OPIECE)'="" F ITEM=1:1:$L($P(@RETURN,U,OPIECE),"  ,") S STR=$P($P(@RETURN,U,OPIECE),",",ITEM) S:$E(STR,1,5)=" and " STR=$E(STR,6,  \*) I STR'="" S ITEMS(STR)="",COUNT=1+$G(COUNT)  ....I $P(REACS(REACT),U,IPIECE)'="" F ITEM=1:1:$L($P(REACS(REACT),U,IPI  ECE),"~") D  .....S STR=$P($P(REACS(REACT),U,IPIECE),"~",ITEM),$P(NEWIVAL,"~",ITEM)=  +STR  .....I STR["-" S STR=$P(STR,"-",2)  .....E S STR=$$EXTERNAL^DILFD(FILE,+FIELD,,STR)  .....I STR'="",'$D(ITEMS(STR)) S ITEMS(STR)="",COUNT=1+$G(COUNT)  ....S ITEM="" F S ITEM=$O(ITEMS(ITEM)) Q:ITEM="" D  .....S:COUNT(1)=COUNT DELIMIT=" and "  .....S TEXT=$S($G(TEXT)'="":TEXT\_DELIMIT,1:"")\_ITEM,COUNT(1)=1+COUNT(1)  ....S $P(@RETURN,U,OPIECE)=$G(TEXT),RETURN=$P(RETURN,"(")  ....I $G(NEWIVAL)'="",NEWIVAL'=$P(@RETURN@(MSGNUM,REACT),U,IPIECE) S $P  (@RETURN@(MSGNUM,REACT),U,IPIECE)=NEWIVAL  ...I FIELD["+" D  ....N FNUM,ITEMS  ....F ITEM=1:1:$L($P(REACS(REACT),U,IPIECE),"~") N STR F FNUM=1:1:$L(F  IELD,"|") D S:STR'="" ITEMS(ITEM)=STR  .....;S STR=$S($G(STR)'="":STR\_U,1:"")\_$$EXTERNAL^DILFD(FILE,+$P(FIELD,  "|",FNUM),,$P($P($P(REACS(REACT),U,IPIECE),"~",ITEM),"|",FNUM))  .....S STR=$S(FNUM>1:STR\_U,1:"")\_$$EXTERNAL^DILFD(FILE,+$P(FIELD,"|",FN  UM),,$P($P($P(REACS(REACT),U,IPIECE),"~",ITEM),"|",FNUM))  ....M:$D(ITEMS)>9 @RETURN=ITEMS  ...I $G(FIELDS1(1.1,1))'="" S FIELDS(1.1,1)=$G(FIELDS1(1.1,1)),FIELDS1(  1.1,1)=""  ...S RETURN=$P(RETURN,"(")  .S @RETURN=MSGNUM(0)  Q FLG |
| ..S TMPFLG=$$PCCHK($$DRP2VACL^GMRAPENC(LPTR))  ..S:TMPFLG>0 FLG=TMPFLG  S:FLG>0 FLG=1  K @RETURN  N STYPE,IPIECE,OPIECE,FIELDS,**FIELDS1,**ITEM,FILE,NODE,FIELD,MSGNUM,RINDEX  ,SITE,SEV  S STYPE("L")="LOCAL",STYPE("R")="REMOTE"  S FIELDS(1,3)="120.8^4^3",FIELDS(1.1,1)="120.85^.01|14.5+^4",FIELDS(2,1  )="120.81^.01\*^5"  S FIELDS(2,2)="120.8^^6",FIELDS(2,3)="120.8^^7"  S FIELDS(2,3,"OTRANSFORM")="S Y=$S(Y["";PS(50.605,"":$P($G(^PS(50.605,+  Y,0)),U,2),1:$P($G(@(U\_$P(Y,"";"",2)\_+Y\_"",0)"")),U)) S:Y="""" Y=""UNKNOWN"""  S FIELDS(1,4)="120.8^6^8"  **S FIELDS1(1.1,1)=""**  I $D(REACS)>9 D  .S REACT="" F S REACT=$O(REACS(REACT)) Q:REACT="" D  **..I $P(REACS(REACT),U,8)="h" S FIELDS1(1.1,1)=$G(FIELDS(1.1,1)),FIELDS(**  **1.1,1)="120.8^9|8+^4"**  ..S REACT(1)=$S($P(REACS(REACT),U,7)'="":$P(REACS(REACT),U,7),1:$P(REAC  S(REACT),U,6))  ..S REACT(2)=$P(REACS(REACT),U,8) S:REACT(2)="" REACT(2)="UNKNOWN"  ..S REACT(3)=$P($P(REACS(REACT),U,4),"|",2) S:REACT(3)="" REACT(3)="UNK  NOWN"  ..I $D(RINDEX(REACT(1),REACT(2),REACT(3))) S MSGNUM=$O(RINDEX(REACT(1),  REACT(2),REACT(3),0))  ..E S MSGNUM(0)=1+$G(MSGNUM(0)),MSGNUM=MSGNUM(0),RINDEX(REACT(1),REACT  (2),REACT(3),MSGNUM)=""  ..S @RETURN@(MSGNUM,REACT)=REACS(REACT),SITE=$P(REACS(REACT),U),@RETURN  @(MSGNUM,"MESSAGE",1)=1+$G(@RETURN@(MSGNUM,"MESSAGE",1))  ..F NODE="ING","CLS","REC" I $D(REACS(REACT,NODE)) F ITEM=1:1:$L(REACS(  REACT,NODE),"~") D  ...S $P(@RETURN@(MSGNUM,"MESSAGE","OFFENDERS",NODE),"~",ITEM)=$P($P(REA  CS(REACT,NODE),"~",ITEM),U)  ...S $P(@RETURN@(MSGNUM,REACT,NODE),"~",ITEM)=$P($P(REACS(REACT,NODE),"  ~",ITEM),U,2)  ..S @RETURN@(MSGNUM,"MESSAGE",1,SITE)=$$GET1^DIQ(4,$P(REACS(REACT),U)\_"  ,",.01)\_U\_STYPE($P(REACS(REACT),U,2))\_U  ..S NODE=0 F S NODE=$O(FIELDS(NODE)) Q:NODE="" S OPIECE=0 F S OPIECE  =$O(FIELDS(NODE,OPIECE)) Q:OPIECE="" D  ...S FILE=$P(FIELDS(NODE,OPIECE),U),FIELD=$P(FIELDS(NODE,OPIECE),U,2),I  PIECE=$P(FIELDS(NODE,OPIECE),U,3)  ...I NODE=1 S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",NODE,SITE))  ...I $L(NODE,".")=2 S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",$P(NODE,".")  ,SITE,$P(NODE,".",2)))  ...I NODE'=1,NODE'["." S RETURN=$NA(@RETURN@(MSGNUM,"MESSAGE",NODE))  ...I FIELD>0,FIELD'["\*",FIELD'["+" S $P(@RETURN,U,OPIECE)=$$EXTERNAL^DI  LFD(FILE,FIELD,,$P(REACS(REACT),U,IPIECE))  ...I 'FIELD,'$D(FIELDS(NODE,OPIECE,"OTRANSFORM")) S $P(@RETURN,U,OPIECE  )=$P(REACS(REACT),U,IPIECE)  ...I $D(FIELDS(NODE,OPIECE,"OTRANSFORM")) N Y S Y=$P(REACS(REACT),U,IPI  ECE) X:Y'="" FIELDS(NODE,OPIECE,"OTRANSFORM") S $P(@RETURN,U,OPIECE)=Y  ...I FIELD["\*" D  ....N TEXT,DELIMIT,ITEMS,STR,COUNT,NEWIVAL  ....S DELIMIT=", ",COUNT=0,COUNT(1)=1  ....I $P($G(@RETURN),U,OPIECE)'="" F ITEM=1:1:$L($P(@RETURN,U,OPIECE),"  ,") S STR=$P($P(@RETURN,U,OPIECE),",",ITEM) S:$E(STR,1,5)=" and " STR=$E(STR,6,  \*) I STR'="" S ITEMS(STR)="",COUNT=1+$G(COUNT)  ....I $P(REACS(REACT),U,IPIECE)'="" F ITEM=1:1:$L($P(REACS(REACT),U,IPI  ECE),"~") D  .....S STR=$P($P(REACS(REACT),U,IPIECE),"~",ITEM),$P(NEWIVAL,"~",ITEM)=  +STR  .....I STR["-" S STR=$P(STR,"-",2)  .....E S STR=$$EXTERNAL^DILFD(FILE,+FIELD,,STR)  .....I STR'="",'$D(ITEMS(STR)) S ITEMS(STR)="",COUNT=1+$G(COUNT)  ....S ITEM="" F S ITEM=$O(ITEMS(ITEM)) Q:ITEM="" D  .....S:COUNT(1)=COUNT DELIMIT=" and "  .....S TEXT=$S($G(TEXT)'="":TEXT\_DELIMIT,1:"")\_ITEM,COUNT(1)=1+COUNT(1)  ....S $P(@RETURN,U,OPIECE)=$G(TEXT),RETURN=$P(RETURN,"(") |
| ....I $G(NEWIVAL)'="",NEWIVAL'=$P(@RETURN@(MSGNUM,REACT),U,IPIECE) S $P  (@RETURN@(MSGNUM,REACT),U,IPIECE)=NEWIVAL  ...I FIELD["+" D  ....N FNUM,ITEMS  ....F ITEM=1:1:$L($P(REACS(REACT),U,IPIECE),"~") N STR F FNUM=1:1:$L(F  IELD,"|") D S:STR'="" ITEMS(ITEM)=STR  **.....;S STR=$S($G(STR)'="":STR\_U,1:"")\_$$EXTERNAL^DILFD(FILE,+$P(FIELD,**  **"|",FNUM),,$P($P($P(REACS(REACT),U,IPIECE),"~",ITEM),"|",FNUM))**  **.....S STR=$S(FNUM>1:STR\_U,1:"")\_$$EXTERNAL^DILFD(FILE,+$P(FIELD,"|",FN**  **UM),,$P($P($P(REACS(REACT),U,IPIECE),"~",ITEM),"|",FNUM))**  ....M:$D(ITEMS)>9 @RETURN=ITEMS  **...I $G(FIELDS1(1.1,1))'="" S FIELDS(1.1,1)=$G(FIELDS1(1.1,1)),FIELDS1(**  **1.1,1)=""**  ...S RETURN=$P(RETURN,"(")  .S @RETURN=MSGNUM(0)  Q FLG  Modified ADDLDATA tag:  ADDLDATA(GLOBAL,PAIEN) ;OBTAIN LOCALLY STORED DATA ELEMENTS FOR ONE REACTION  ;PARAMETERS: GLOBAL => GLOBAL REFERENCE WHERE TO RETURN THE DATA  ; PAIEN => IEN IN FILE #120.8  N SITE  S SITE=$$SITE^VASITE(),SITE=$P(SITE,U)\_U\_"L"\_U  S @GLOBAL=SITE\_$P(^GMR(120.8,PAIEN,0),U,4)  N REACT,IEN  S REACT=0 F S REACT=$O(^GMR(120.8,PAIEN,10,REACT)) Q:+$G(REACT)=0 D  .S REACT("VALUE")=$P($G(^GMR(120.8,PAIEN,10,REACT,0)),U)  .I REACT("VALUE")=GMRAOTH D  ..S REACT("VALUE")=REACT("VALUE")\_"-"\_$P($G(^GMR(120.8,PAIEN,10,REACT,0  )),U,2)  ..I $P(REACT("VALUE"),"-",2)="" D  ...I GMRAOTH>0 S $P(REACT("VALUE"),"-",2)=$P($G(^GMRD(120.83,GMRAOTH,0)  ),U)  ...E K REACT("VALUE")  .S:$D(REACT("VALUE")) $P(@GLOBAL,U,5)=$S($P($G(@GLOBAL),U,5)'="":$P(@GL  OBAL,U,5)\_"~",1:"")\_REACT("VALUE")  S $P(@GLOBAL,U,6)=$P(^GMR(120.8,PAIEN,0),U,2)  S $P(@GLOBAL,U,7)=$P(^GMR(120.8,PAIEN,0),U,3)  S $P(@GLOBAL,U,8)=$P(^GMR(120.8,PAIEN,0),U,6)  S $P(@GLOBAL,U,9)="V"  **I $P(@GLOBAL,U,8)="h",+$P($G(^GMR(120.8,PAIEN,0)),U,8)>0 D**  **. S $P(@GLOBAL,U,4)=$S($P($G(@GLOBAL),U,4)'="":$P(@GLOBAL,U,4)\_"~",1:""**  **)\_$P($G(^GMR(120.8,PAIEN,0)),U,9)\_"|"\_$P($G(^(0)),U,8)**  Q:$D(^GMR(120.85,"C",PAIEN))<10  S IEN=0 F S IEN=$O(^GMR(120.85,"C",PAIEN,IEN)) Q:'+IEN D  .S $P(@GLOBAL,U,4)=$S($P($G(@GLOBAL),U,4)'="":$P(@GLOBAL,U,4)\_"~",1:"")  \_$P($G(^GMR(120.85,IEN,0)),U)\_"|"\_$P($G(^(0)),U,14)  Q |

###### Routine GMRAOR1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAOR1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified EN1 tag:**  EN1(DFN,ARRAY) ;RETURN CONDENSED LIST OF PATIENT REACTIONS  ;CONTROLLED BY CONTROLLED SUBSCRIPTION INTEGRATION AGREEMENT #2421  ;PARAMETERS: DFN => INTERNAL ENTRY NUMBER (IEN) OF THE PATIENT  ; IN THE PATIENT FILE (#2).  ; ARRAY => NAME OF THE ARRAY IN WHICH TO RETURN THE  ; REACTIONS. IF NOTHING IS PASSED IN, DATA  ; IS RETURNED IN THE GMRARXN ARRAY.  ; GMRA => REACTION SEARCH CRITERIA; REFER TO ^GMRADPT  ; FOR DETAILS  ; GMRAIDT => WHEN SET, WILL RETURN THE DATE ENTERED FOR  ; THE SIGN/SYMPTOM IN THE SECOND SEMI-COLON  ; PIECE  Q:+$G(DFN)'>0  S ARRAY=$G(ARRAY,"GMRARXN")  Q:ARRAY="GMRAL"  N GMRAL  K @ARRAY  D EN2^GMRADPT  I GMRAL D  .N GMRAIEN,GMRASS  .S GMRAIEN="" F S GMRAIEN=$O(GMRAL(GMRAIEN)) Q:GMRAIEN="" D  ..N GMRAGMR,GMRAIDX,GMRASEV  ..S @ARRAY=1+$G(@ARRAY)  ..I $P(GMRAL(GMRAIEN),U,7)["D" D  ...S GMRAGMR=$$EXTERNAL^DILFD(120.8,1,,$P(GMRAL(GMRAIEN),U,9))  ...I $P(GMRAL(GMRAIEN),U,2)'=GMRAGMR,($P(GMRAL(GMRAIEN),U,9)'["50.605")  S @ARRAY@(@ARRAY)=GMRAGMR\_" ["\_$P(GMRAL(GMRAIEN),U,2)\_"]"  ..I $P(GMRAL(GMRAIEN),U,7)'["D"!('$D(@ARRAY@(@ARRAY))) D  ...S @ARRAY@(@ARRAY)=$P(GMRAL(GMRAIEN),U,2)  **..I $P($P(GMRAL(GMRAIEN),U,10),";",2)="o" D**  **...S GMRAIDX=0 F S GMRAIDX=$O(GMRAL(GMRAIEN,"O",GMRAIDX)) Q:'+GMRAIDX**  **D**  **....I $P($P(GMRAL(GMRAIEN,"O",GMRAIDX),U),";",2)>$G(GMRASEV) S GMRASEV=**  **$P($P(GMRAL(GMRAIEN,"O",GMRAIDX),U),";",2)**  **...I $G(GMRASEV)>0 S GMRASEV=$$EXTERNAL^DILFD(120.85,14.5,,GMRASEV)**  **..I $P($P(GMRAL(GMRAIEN),U,10),";",2)="h" D**  **...S GMRASEV=$P($P(GMRAL(GMRAIEN,"H"),U),";")**  ..S @ARRAY@(@ARRAY)=@ARRAY@(@ARRAY)\_U\_$G(GMRASEV)\_U\_GMRAIEN  ..S GMRASS=0 F S GMRASS=$O(GMRAL(GMRAIEN,"S",GMRASS)) Q:GMRASS<1 D  ...S @ARRAY@(@ARRAY,"S",GMRASS)=$P($P(GMRAL(GMRAIEN,"S",GMRASS),U),";")  \_$S($G(GMRAIDT):";"\_$P($P(GMRAL(GMRAIEN,"S",GMRASS),U,2),";",2),1:"")  S @ARRAY=GMRAL  Q |

###### Routine GMRAOR2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAOR2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified DATA tag:**  DATA(IEN,ARRAY) ;ASSEMBLE DATA TO RETURN  ;PARAMETERS: IEN => IEN IN FILE #120.8  ; ARRAY => NAME OF THE ARRAY IN WHICH TO RETURN THE  ; REACTION DATA.  Q:$G(IEN)=""  N GMRAPA,GMRAOTH,GMRAI,GMRAGMR,GMRAORIG,GMRAIDX  K @ARRAY  S GMRAPA=IEN,GMRAPA(0)=$G(^GMR(120.8,GMRAPA,0)) Q:GMRAPA(0)=""  ;\*BD  I $G(GMRAV1) S @ARRAY=$P(GMRAPA(0),U,2)\_U  I '$G(GMRAV1) D  .;\*ED (C U DOTS)  .I $P(GMRAPA(0),U,20)["D" D  ..S GMRAGMR=$$EXTERNAL^DILFD(120.8,1,,$P(GMRAPA(0),U,3))  ..I $P(GMRAPA(0),U,2)'=GMRAGMR S @ARRAY=GMRAGMR\_" ["\_$P(GMRAPA(0),U,2)\_  "]"\_U  .I $P(GMRAPA(0),U,20)'["D"!('$D(@ARRAY)) D  ..S @ARRAY=$P(GMRAPA(0),U,2)\_U  S GMRAORIG=$P(GMRAPA(0),U,5)  I +GMRAORIG>0 D  .S @ARRAY=@ARRAY\_$$GET1^DIQ(120.8,GMRAPA\_",",5)\_U\_$$GET1^DIQ(200,GMRAOR  IG\_",",8)\_U  I +GMRAORIG=0 D  .S @ARRAY=@ARRAY\_"<None>"\_U\_U  S @ARRAY=@ARRAY\_$S($P(GMRAPA(0),U,16)=1:"",1:"NOT ")\_"VERIFIED"\_U  S @ARRAY=@ARRAY\_$S($P(GMRAPA(0),U,6)="o":"OBSERVED",$P(GMRAPA(0),U,6)="  h":"HISTORICAL",1:"")\_U  S @ARRAY=@ARRAY\_$S($P(GMRAPA(0),U,14)="A":"ALLERGY",$P(GMRAPA(0),U,14)=  "P":"PHARMACOLOGIC",$P(GMRAPA(0),U,14)="U":"UNKNOWN",1:"")\_U  S @ARRAY=@ARRAY\_$$OUTTYPE^GMRAUTL($P(GMRAPA(0),U,20))\_U\_$S($P(GMRAPA(0)  ,U,16)&('$P(GMRAPA(0),U,18)):"<auto-verified>",1:$$GET1^DIQ(120.8,GMRAPA\_",",21)  )\_U\_$P(GMRAPA(0),U,17)  S @ARRAY=@ARRAY\_U\_$$FMTE^XLFDT($P(GMRAPA(0),U,4))  ;Comments  S GMRAI=0 F GMRAIDX=1:1 S GMRAI=$O(^GMR(120.8,GMRAPA,26,GMRAI)) Q:GMRAI  <1 D  .N GMRACOM  .S GMRACOM=$G(^GMR(120.8,GMRAPA,26,GMRAI,0)) Q:GMRACOM=""  .S @ARRAY@("C",GMRAIDX)=$P(GMRACOM,U)\_U\_$S($P(GMRACOM,U,3)="V":"VERIFIE  R",$P(GMRACOM,U,3)="O":"ORIGINATOR",1:"")\_U\_$$GET1^DIQ(200,$P(GMRACOM,U,2)\_",",.  01)  .M @ARRAY@("C",GMRAIDX)=^GMR(120.8,GMRAPA,26,GMRAI,2)  ;Observer information from file #120.85  **I $P(GMRAPA(0),U,6)="o" D**  **.S GMRAI=0 F GMRAIDX=1:1 S GMRAI=$O(^GMR(120.85,"C",GMRAPA,GMRAI)) Q:GM**  **RAI<1 D**  **..N GMRACOM**  **..S GMRACOM=$G(^GMR(120.85,GMRAI,0)) Q:GMRACOM=""**  **..S @ARRAY@("O",GMRAIDX)=$P(GMRACOM,U)\_U\_$S($P(GMRACOM,U,14)=1:"MILD",$**  **P(GMRACOM,U,14)=2:"MODERATE",$P(GMRACOM,U,14)=3:"SEVERE",1:"")**  **;Historical information**  **I $P(GMRAPA(0),U,6)="h" D**  **.I $P(GMRAPA(0),U,8)="",$P(GMRAPA(0),U,9)="" Q**  **.S @ARRAY@("H")=$P(GMRAPA(0),U,9)\_U\_$S($P(GMRAPA(0),U,8)=1:"MILD",$P(GM**  **RAPA(0),U,8)=2:"MODERATE",$P(GMRAPA(0),U,8)=3:"SEVERE",1:"")**  ;Signs/Symptoms  S GMRAOTH=$O(^GMRD(120.83,"B","OTHER REACTION",0))  S GMRAI=0 F GMRAIDX=1:1 S GMRAI=$O(^GMR(120.8,GMRAPA,10,GMRAI)) Q:GMRAI  <1 D  .N GMRAZ  .S GMRAZ=$G(^GMR(120.8,GMRAPA,10,GMRAI,0)) Q:GMRAZ=""  .S @ARRAY@("S",GMRAIDX)=$S(+GMRAZ'=GMRAOTH:$P($G(^GMRD(120.83,+GMRAZ,0)  ),U),1:$P(GMRAZ,U,2))\_$S($P(GMRAZ,U,4)'="":" ("\_$$FMTE^XLFDT($P(GMRAZ,U,4),2)\_")  ",1:"")  ;VA Drug Classes  S GMRAI=0 F GMRAIDX=1:1 S GMRAI=$O(^GMR(120.8,GMRAPA,3,GMRAI)) Q:GMRAI<  1 D  .N GMRACOM |
| .S GMRACOM=$G(^GMR(120.8,GMRAPA,3,GMRAI,0)) Q:GMRACOM=""  .S @ARRAY@("V",GMRAIDX)=$$CLP2CLDA^GMRAPENC(GMRACOM)  ;Drug Ingredients  S GMRAI=0 F GMRAIDX=1:1 S GMRAI=$O(^GMR(120.8,GMRAPA,2,GMRAI)) Q:GMRAI<  1 D  .N GMRACOM  .S GMRACOM=$G(^GMR(120.8,GMRAPA,2,GMRAI,0)) Q:GMRACOM=""  .S @ARRAY@("I",GMRAIDX)=$$INP2INNA^GMRAPENC(GMRACOM)  Q |

###### Routine GMRAPEH0

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAPEH0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified:**  EN1 ; ENTRY FROM MAIN A/AR EDIT TO EDIT HISTORICAL A/AR  **N GMRAREQ,GMRAREQS**  **S DIE="^GMR(120.8,"**  **S DA=GMRAPA**  **S DR="9;8"**  **D ^DIE**  **I $D(DTOUT)!$D(DUOUT) G EXIT**  **K DA,DIE,DR,X,Y**  **SIG S GMRAREQS=0**  **D EN1^GMRAPER2(GMRAPA,"120.8",.GMRAOUT,$P(^GMR(120.8,GMRAPA,0),U,9)) G:**  **GMRAOUT EXIT**  I $D(^XUSEC("GMRA-ALLERGY VERIFY",DUZ)) D MECH^GMRAPED0  G EXIT:GMRAOUT  COM S GMRAVCM="O" D ENDING^GMRAPEM1  I $D(DTOUT)!$D(Y) S GMRAOUT=1  I 'GMRAOUT D COMCHECK  I 'GMRAOUT G:GMRAREQ COM  **I 'GMRAOUT,$P(GMRAPA(0),"^",6)="h" D CKSIGCOM**  **I 'GMRAOUT G:GMRAREQS SIG**  S GMRAOUT=0  K DA,DR,DIE,DUOUT,DTOUT  Q  **Added CKSIGCOM:**  **CKSIGCOM ; Check to require Historical entry to have**  **; a sign/symptom entered or a minimum of 4 characters**  **; in a comment.**  **N CHARCNT,GMRAX,X**  **S GMRAREQS=0**  **I +DA>0 D**  **. S (X,CHARCNT)=0**  **. F S X=$O(^GMR(120.8,GMRAPA,26,DA,2,X)) Q:X="" D Q:CHARCNT>3**  **.. S CHARCNT=CHARCNT+$L($G(^GMR(120.8,GMRAPA,26,DA,2,X,0)))**  **S GMRAX=$O(^GMR(120.8,GMRAPA,10,0))**  **S GMRAREQS=$S(((+CHARCNT<4)&(+GMRAX<1)):1,1:0)**  **I 'GMRAREQS Q**  **W !!,$C(7),"AT LEAST ONE SIGN/SYMPTOM OR A COMMENT, OF AT LEAST FOUR"**  **W !,$C(7),"CHARACTERS, MUST BE ENTERED FOR HISTORICAL REACTIONS." D HAN**  **GT**  **I $D(DIRUT) S GMRAOUT=1**  **K DIRUT**  **Q** |

###### Routine GMRAPEM1

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAPEM1 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified EDIT tag:**  EDIT ;Edit the word processing field.  Q:GMRAOUT=2 S:'$D(GMRAVCM) GMRAVCM="O"  S GMRAOUT=0 N GMRA  I '$D(^GMR(120.8,GMRAPA,26,0)) S ^(0)="^120.826D^^"  D NOW^%DTC S DIC="^GMR(120.8,"\_GMRAPA\_",26,",DA(1)=GMRAPA,DLAYGO=120.8,  DIC(0)="L",X=% K DD,DO,DINUM D FILE^DICN K DLAYGO G EXIT:+Y'>0  K DR S DA=+Y,DIE=DIC,DR="1////"\_DUZ\_";1.5////"\_GMRAVCM\_";2" K DIC D ^DI  E  I '$O(^GMR(120.8,DA(1),26,DA,2,0)) S DIK=DIE D ^DIK **K Y** |

###### Routine GMRAPER0

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | GMRAPER0 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified DATE tag:**  DATE(DATE,ASK) ; Enter the date for a reaction  Q:ASK  N %DT,X,Y  S DATE=$G(DATE,""),%DT="AEPT",%DT("A")="Date(Time Optional) of appearan  ce of Sign/Symptom(s): "  S:$P(GMRAPA(0),U,6)="o" %DT("B")=$S(DATE="":"NOW",1:$$FMTE^XLFDT(DATE,1  ))  **S:$P(GMRAPA(0),U,6)="h" %DT("B")=$S(DATE="":"",1:$$FMTE^XLFDT(DATE,1))**  S %DT(0)="-NOW" D ^%DT I "^^"[X S GMRAOUT=$L(X) Q  S DATE=Y,ASK=1  Q |

###### Routine ORQQAL

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call |  |

| 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 |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified DETAIL:**  DETAIL(ORAY,DFN,ALLR,ID) ; RETURN DETAILED ALLERGY INFO FOR SPECIFIED ALLERGIC R  EACTION:  D EN2^GMRAOR2(ALLR,"GMRACT")  N CR,OX,OH S CR=$CHAR(13),I=1  S ORAY(I)=" Causative agent: "\_$P(GMRACT,U),I=I+1  S ORAY(I)=" Nature of Reaction: "\_$S($P(GMRACT,U,6)="ALLERGY":"Allergy"  ,$P(GMRACT,U,6)="PHARMACOLOGIC":"Adverse Reaction",$P(GMRACT,U,6)="UNKNOWN":"Unk  nown",1:""),I=I+1 ;216  S ORAY(I)=" ",I=I+1  I $D(GMRACT("S",1)) D SYMP  I $D(GMRACT("V",1)) D CLAS  S ORAY(I)=" Originator: "\_$P(GMRACT,U,2)\_$S($L($P(GMRACT,U,3)):  " ("\_$P(GMRACT,U,3)\_")",1:""),I=I+1 ;216  S ORAY(I)=" Originated: "\_$P(GMRACT,U,10),I=I+1 ;216  I $D(GMRACT("O",1)) D OBS  **I $D(GMRACT("H")) D HIST**  S ORAY(I)=" Verified: "\_$S($P(GMRACT,U,4)="VERIFIED":$P(GMRAC  T,U,8),1:"No"),I=I+1 ;216  S ORAY(I)="Observed/Historical: "\_$S($P(GMRACT,U,5)="OBSERVED":"Observe  d",$P(GMRACT,U,5)="HISTORICAL":"Historical",1:""),I=I+1  I $D(GMRACT("C",1)) D COM  K GMRACT  Q  **Added HIST process:**  **HIST ;Historical Severity and dates**  **N Y**  **S Y=$P(GMRACT("H"),U) D DD^%DT**  **S ORAY(I)=" Hist date/severity: "\_$S(Y="":"",1:Y\_" ")\_$P(GMRACT("H"),U,**  **2),I=I+1**  **S ORAY(I)=" ",I=I+1**  **Q** |

##### NSR # 20070817 – Progress Notes Display Misleading (RSD #2.6.25)

###### Routine ORCNOTE

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call | HASDAD^TIUSRVLI |

| 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: DFN  Definition: Patient Internal Entry Number | | | | |
| **Output Attribute Name and Definition** | Name: ORY  Definition: Total count of all signed notes, minus addendums and child interdisciplinary notes. | | | | |

| Current Logic |
| --- |
| **Original GETTOT tag:**  GETTOT(ORY,DFN) ; --get total count of progress notes  ; ^TIU(8925,"ACLPT",3) refers to progress notes class  N ORTIDT,ORTIFN  S ORY=0  Q:'$G(DFN)  F ORTIDT=0:0 S ORTIDT=$O(^TIU(8925,"ACLPT",3,DFN,ORTIDT)) Q:ORTIDT<1 D  .F ORTIFN=0:0 S ORTIFN=$O(^TIU(8925,"ACLPT",3,DFN,ORTIDT,ORTIFN)) Q:ORT  IFN<1 S ORY=ORY+1  Q |

| Modified Logic (Changes are in bold) |
| --- |
| **Modified GETTOT tag:**  GETTOT(ORY,DFN) ; --get total count of signed progress notes**, not**  **; counting addendums and child interdisciplinary notes**  ; ^TIU(8925,"ACLPT",3) refers to progress notes class  N ORTIDT,ORTIFN  S ORY=0  Q:'$G(DFN)  F ORTIDT=0:0 S ORTIDT=$O(^TIU(8925,"ACLPT",3,DFN,ORTIDT)) Q:ORTIDT<1 D  .F ORTIFN=0:0 S ORTIFN=$O(^TIU(8925,"ACLPT",3,DFN,ORTIDT,ORTIFN)) Q:ORTIFN<1 **D**  **..I +$$HASDAD^TIUSRVLI(ORTIFN) Q**  ..S ORY=ORY+1  **..Q**  Q |

###### Routine TIUSRVLI

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

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

| 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: FLG  Definition: Flag to enable the removal of notes from the return global if they have been returned previously. Used to identify the originating call from ACLPT^TIUSRVLO. | | | | |
| **Output Attribute Name and Definition** | Name: TIUY  Definition: Reference to the global array which results are returned in  Name: TIUI  Definition: A counter used for setting entries into the TIUY reference. | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| TIUSRVLI ; SLC/JER - Server fns - lists for CPRS ;03/30/17 15:20  ;;1.0;TEXT INTEGRATION UTILITIES;\*\*108,122,251**,289**\*\*;Jun 20, 1997  HASDAD(DA) ; Evaluate whether a document has a parent  Q $S(+$P($G(^TIU(8925,+DA,0)),U,6):1,+$G(^TIU(8925,+DA,21)):1,1:0)  ;  SETDAD(TIUY,DA,TIUI**,FLG**) ; Set parent in return array  N DADA**,RMVFLG**,TIUD0,TIUD21  **S FLG=+$G(FLG)**  ; Exclude components  Q:'+$$ISDOC(DA)  S TIUD0=$G(^TIU(8925,DA,0)),TIUD21=$G(^(21))  S DADA=$S(+$P(TIUD0,U,6):+$P(TIUD0,U,6),+TIUD21:+TIUD21,1:0)  Q:+DADA'>0  Q:+$D(@TIUY@("INDX",DADA))  Q:+$D(^TIU(8925,DADA,0))=0  **S RMVFLG=0**  **I FLG S RMVFLG=$$DOREMOV(.TIUY,DADA)**  **Q:RMVFLG=1**  S TIUI=$S($G(SEQUENCE)="A":+$G(TIUI)-1,1:+$G(TIUI)+1)  S @TIUY@(TIUI)=DADA\_U\_$$RESOLVE^TIUSRVLO(DADA)  S @TIUY@("INDX",DADA,TIUI)=""  I +$G(SHOWADD) D SETKIDS(.TIUY,DADA,.TIUI**,FLG**)  I +$$HASDAD(DADA) D SETDAD(.TIUY,DADA,.TIUI**,FLG**)  Q  ;  HASKIDS(DA) ; Evaluate whether a document has children  N TIUY,KIDA S (KIDA,TIUY)=0  ; Check for addenda  F S KIDA=$O(^TIU(8925,"DAD",DA,KIDA)) Q:+TIUY!(+KIDA'>0) D  . I '+$$ISCOMP^TIUSRVR1(KIDA) S TIUY=1  I +TIUY G HASKIDX  ; Next, look for ID Entries  S TIUY=$S(+$O(^TIU(8925,"GDAD",DA,0)):1,1:0)  HASKIDX Q TIUY  ;  SETKIDS(TIUY,DA,TIUI**,FLG**) ; Set children in return array  N KIDA**,RMVFLG**  **S FLG=+$G(FLG)**  **S RMVFLG=0**  S KIDA=0  ; Begin with addenda  F S KIDA=$O(^TIU(8925,"DAD",DA,KIDA)) Q:+KIDA'>0 D **Q:RMVFLG=1**  . Q:'+$$ISDOC(KIDA)  . Q:+$D(@TIUY@("INDX",KIDA))  **. I FLG S RMVFLG=$$DOREMOV(.TIUY,KIDA)**  **. Q:RMVFLG=1**  . S TIUI=$S($G(SEQUENCE)="A":+$G(TIUI)-1,1:+$G(TIUI)+1)  . S @TIUY@(TIUI)=KIDA\_U\_$$RESOLVE^TIUSRVLO(KIDA)  . S @TIUY@("INDX",KIDA,TIUI)=""  ; Next do ID entries  S KIDA=0  F S KIDA=$O(^TIU(8925,"GDAD",DA,KIDA)) Q:+KIDA'>0 D **Q:RMVFLG=1**  . Q:+$D(@TIUY@("INDX",KIDA))  **. I FLG S RMVFLG=$$DOREMOV(.TIUY,KIDA)**  **. Q:RMVFLG=1** |
| . S TIUI=$S(SEQUENCE="A":+$G(TIUI)-1,1:+$G(TIUI)+1)  . S @TIUY@(TIUI)=KIDA\_U\_$$RESOLVE^TIUSRVLO(KIDA)  . S @TIUY@("INDX",KIDA,TIUI)=""  . I +$$HASKIDS(KIDA) D SETKIDS(.TIUY,KIDA,.TIUI**,FLG**)  Q  ISDOC(DA) ; Evaluate whether a given record is a document  N TIUY,TIUTYP  S TIUTYP=+$G(^TIU(8925,DA,0))  S TIUY=$S($P($G(^TIU(8925.1,+TIUTYP,0)),U,4)="DOC":1,1:0)  Q TIUY  GETUND(TIUY,CLASS,DFN,TIME1,TIME2,TIUJ,SEQUENCE) ; Get undictated docs  N TIUTYP,TIUI,DATTIM  D DOCTYPE^TIUSRVL(.TIUTYP,CLASS) Q:+$D(TIUTYP)'>9  S TIUI=0  F S TIUI=$O(TIUTYP(TIUI)) Q:+TIUI'>0 D  . N STATUS  . F STATUS=1:1:2 D  . . S DATTIM=TIME1-.0000001  . . F S DATTIM=$O(^TIU(8925,"APT",DFN,+TIUTYP(TIUI),STATUS,DATTIM)) Q:+DATTIM'>0 D  . . . N TIUDA S TIUDA=0  . . . F S TIUDA=$O(^TIU(8925,"APT",DFN,+TIUTYP(TIUI),STATUS,DATTIM,TIUDA)) Q:+TIUDA'>0 D  . . . . Q:+$D(@TIUY@("INDX",TIUDA))  . . . . S TIUJ=$S(SEQUENCE="A":+$G(TIUJ)-1,1:+$G(TIUJ)+1)  . . . . S @TIUY@(TIUJ)=TIUDA\_U\_$$RESOLVE^TIUSRVLO(TIUDA)  . . . . S @TIUY@("INDX",TIUDA,TIUJ)=""  Q  **DOREMOV(TIUY,DA) ;Check for need to remove previously returned data**  **; and perform the remove**  **N RMVFLG,TIUD1301**  **S RMVFLG=0**  **S TIUD1301=$P($G(^TIU(8925,DA,13)),U,1)**  **I +$G(PRVRTNDT)>0 D**  **. I PRVRTNDT=TIUD1301,TIUIEN>DA Q**  **. I PRVRTNDT>TIUD1301 Q**  **. D RMV(.TIUY,DA)**  **. S RMVFLG=1**  **. Q**  **Q RMVFLG**  **RMV(TIUY,DA) ; Remove notes previously returned**  **D RMVDAD(.TIUY,DA)**  **D RMVKID(.TIUY,DA)**  **I TIUJ>0 S TIUJ=TIUJ-1**  **Q**  **RMVDAD(TIUY,DA) ; Remove parent entries**  **N CNT,DADA,TIUD0,TIUD21**  **; Exclude components**  **Q:'+$$ISDOC(DA)**  **S TIUD0=$G(^TIU(8925,DA,0)),TIUD21=$G(^(21))**  **S DADA=$S(+$P(TIUD0,U,6):+$P(TIUD0,U,6),+TIUD21:+TIUD21,1:0)**  **Q:+DADA'>0**  **S @TIUY@("INDX",DADA)=""**  **S CNT=+$O(@TIUY@("INDX",DADA,""))** |
| **Q:CNT<1**  **K @TIUY@(CNT)**  **I +$G(SHOWADD) D RMVKID(.TIUY,DADA)**  **I +$$HASDAD(DADA) D RMVDAD(.TIUY,DADA)**  **Q**  **RMVKID(TIUY,DA) ; Remove child notes**  **N CNT,KIDA**  **S KIDA=0**  **; Begin with addenda**  **F S KIDA=$O(^TIU(8925,"DAD",DA,KIDA)) Q:+KIDA'>0 D**  **. Q:'+$$ISDOC(KIDA)**  **. S @TIUY@("INDX",KIDA)=""**  **. S CNT=+$O(@TIUY@("INDX",KIDA,""))**  **. Q:CNT<1**  **. K @TIUY@(CNT)**  **. Q**  **; Next do ID entries**  **S KIDA=0**  **F S KIDA=$O(^TIU(8925,"GDAD",DA,KIDA)) Q:+KIDA'>0 D**  **. S @TIUY@("INDX",KIDA)=""**  **. S CNT=+$O(@TIUY@("INDX",KIDA,""))**  **. Q:CNT<1**  **. K @TIUY@(CNT)**  **. I +$$HASKIDS(KIDA) D RMVKID(.TIUY,KIDA)**  **. Q**  **Q**  **RESEQ(TIUY,TIUI) ; Resequence @TIUY@(TIUI) after removal of entries and reset TIUI**  **N ADD,ARRY,CNT,CNT1,DA,DATA**  **S ADD=$S(SEQUENCE="D":1,0:-1)**  **S (CNT,CNT1)=0**  **F S CNT1=$O(@TIUY@(CNT1)) Q:+CNT1=0 D**  **. S CNT=CNT+ADD**  **. S ARRY(CNT)=$G(@TIUY@(CNT1))**  **. Q**  **K @TIUY**  **S (CNT,TIUI)=0**  **F S CNT=$O(ARRY(CNT)) Q:+CNT=0 D**  **. S DATA=$G(ARRY(CNT))**  **. S DA=$P(DATA,U,1)**  **. S @TIUY@(CNT)=DATA**  **. S @TIUY@("INDEX",DA,CNT)=""**  **. S TIUI=CNT**  **. Q**  **Q** |

###### Routine TIUSRVLO

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

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call | TIUSRVLI |

| 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: SHOW  Definition: Boolean value which tells the system to return "0^SHOW MORE" in the return array when additional notes are available for context of 1 or 5 when occurrence limit prevents all notes from displaying.  Name: TIUIEN  Definition: Starting TIU IEN for finding additional return values when "SHOW MORE" was returned in the previous return array. | | | | |
| **Output Attribute Name and Definition** | Name:  Definition: | | | | |

| Current Logic |
| --- |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| TIUSRVLO ; SLC/JER - Server fns - lists for CPRS ;03/30/17 14:13  ;;1.0;TEXT INTEGRATION UTILITIES;\*\*1,15,19,63,108,122,181,194,211,268,**289**\*\*;Jun 20, 1997;Build 47  **Modified CONTEXT tag:**  CONTEXT(TIUY,CLASS,CONTEXT,DFN,EARLY,LATE,PERSON,OCCLIM,SEQUENCE,SHOWADD,INCUND,**SHOW,TIUIEN**) ; main  ; --- Call with: TIUY - RETURN ARRAY pass by reference  ; CLASS - Pointer to TIU DOCUMENT DEFINITION #8925.1  ; CONTEXT - 1=All Signed (by PT),  ; - 2="Unsigned (by PT&(AUTHOR!TANSCRIBER))  ; - 3="Uncosigned (by PT&EXPECTED COSIGNER  ; - 4="Signed notes (by PT&selected author)  ; - 5="Signed notes (by PT&date range)  ; DFN - Pointer to Patient (#2)  ; [EARLY] - FM date/time to begin search  ; [LATE] - FM date/time to end search  ; [PERSON] - Pointer to file 200 (DUZ if not passed)  ; [OCCLIM] - Occurrence Limit (optional)  ; [SEQUENCE]- "A"=ascending (Regular date/time)  ; - "D"=descending (Reverse date/time) (dflt)  ; [INCUND] - Boolean: include undictated & untranscribed  **; SHOW - Boolean: Return "0^SHOW MORE" in return**  **; array when additional notes available for**  **; context of 1 or 5 when occurrence limit**  **; prevents all notes from displaying**  **; TIUIEN - Starting TIU IEN for additional return**  **; when "SHOW MORE" was received in previous**  **; return array (LATE date/time will be set**  **; to Reference date of this TIU document)**  ;  S TIUY=$NA(^TMP("TIUR",$J))  K @TIUY  I $G(CONTEXT)'>0 Q  I $G(CLASS)'>0 Q  I $G(CONTEXT)=1 D STRT1^AWCMCPR1 ; TIU\*1.0\*181  S:+$G(EARLY)'>0!(+$G(CONTEXT)=1) EARLY=0  S:+$G(LATE)'>0!(+$G(CONTEXT)=1) LATE=5000000  I EARLY>LATE D SWAP(.EARLY,.LATE)  **I +$G(TIUIEN)>0,(CONTEXT=1!(CONTEXT=5)) S LATE=$P($G(^TIU(8925,+TIUIEN,13)),U,1) I EARLY>LATE Q**  I $L(LATE,".")=1 D EXPRANGE(.EARLY,.LATE)  S:+$G(PERSON)'>0 PERSON=DUZ  S:$G(SEQUENCE)']"" SEQUENCE="D"  S:+$G(OCCLIM)'>0 OCCLIM=9999999  S DFN=+$G(DFN)  S EARLY=9999999-EARLY,LATE=9999999-LATE ; CHANGE TO REVERSE DATES  ;  I CONTEXT=1!(CONTEXT=5) D G CTXQ  . D ACLPT(.TIUY,CLASS,DFN,LATE,EARLY,OCCLIM,SEQUENCE**,+$G(SHOW),$G(TIUIEN)**)  ;  I CONTEXT=2 D G CTXQ  . I DFN>0 D Q  . . D ACLAU(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE,$G(INCUND))  . F S DFN=$O(^TIU(8925,"ACLAU",CLASS,PERSON,DFN)) Q:DFN'>0 D ACLAU(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE,$G(INCUND))  ;  I CONTEXT=3 D G CTXQ  . I DFN>0 D Q |
| . . D ACLEC(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE)  . F S DFN=$O(^TIU(8925,"ACLEC",CLASS,PERSON,DFN)) Q:DFN'>0 D ACLEC(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE)  ;  I CONTEXT=4 D G CTXQ  . I DFN>0 D Q  . . ;VMP OIFO BAY PINES;ELR;TIU\*1.0\*194 REMOVED EXECUTION OF ACLSB & ADDED APTCL  . . ;D ACLSB(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE)  . . D APTCL^TIUSRVLP(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE)  . F S DFN=$O(^TIU(8925,"APTCL",DFN)) Q:DFN'>0 D APTCL^TIUSRVLP(.TIUY,CLASS,PERSON,DFN,LATE,EARLY,SEQUENCE)  ;  **Modified ACLPT, ACLAU, ACLEC, ACLSB tags:**  ACLPT(TIUY,CLASS,DFN,TIME1,TIME2,OCCLIM,SEQUENCE**,SHOW,TIUIEN**) ; Signed, by patient  N DATTIM,DA,**LSTDA,NGD,PRVRTNDT,**ROOT,**STRTDT,**TIUI,TIUJ,TIUP  **S (LSTDA,NGD,STRTDT)=0**  **I +TIUIEN>0 D**  **. S NGD=1**  **. S STRTDT=9999999-TIME1**  **. S PRVRTNDT=0**  S ROOT=$NA(^TIU(8925,"ACLPT",CLASS,DFN)),TIUJ=0  S DATTIM=TIME1-.0000001  F S DATTIM=$O(@ROOT@(DATTIM)) Q:$S(+DATTIM'>0:1,+DATTIM>TIME2:1,+$G(TIUJ)'<OCCLIM:1,1:0) D  . S DA=0 F S DA=$O(@ROOT@(DATTIM,DA)) Q:DA'>0 **Q:+$G(TIUJ)'<OCCLIM** D  **. . I NGD=1 D Q ;If grabbing additional entries, don't return any before the one sent in or the sent in one as the starting point**  **. . . I DA=TIUIEN D ;Reached starting point, everything after is valid to return**  **. . . . S NGD=0**  **. . . . S PRVRTNDT=9999999-DATTIM**  **. . . . Q**  **. . . Q**  . . ;Added first condition to following line for interdisciplinary notes - need addenda  . . I +$G(SHOWADD)=0,(+$G(^TIU(8925,+DA,0))=81) Q  . . I +$G(^TIU(8925,+DA,0))'>0 K @ROOT@(DATTIM,DA) Q  . . Q:+$D(@TIUY@("INDX",DA))  . . ;Q:'+$$ISDOC^TIUSRVLI(DA) ;If record not a document Quit  . . ; Selectively filter DELETED or RETRACTED records  . . I +$P($G(^TIU(8925,DA,0)),U,5)>13,'+$$CANDO^TIULP(DA,"VIEW",DUZ) Q  . . S TIUI=$S(SEQUENCE="D":+$G(TIUI)+1,1:+$G(TIUI)-1)  . . S @TIUY@(TIUI)=DA\_U\_$$RESOLVE(DA)  **. . I '+$D(@TIUY@("INDX",+$P($G(^TIU(8925,+DA,0)),U,6))),'+$D(@TIUY@("INDX",+$G(^TIU(8925,+DA,21**)))) S TIUJ=+$G(TIUJ)+1  . . **;**S TIUJ=+$G(TIUJ)+1  . . S @TIUY@("INDX",DA,TIUI)=""  **. . S LSTDA=DA**  . . Q:+$G(SHOWADD)=0  . . S TIUP=+$$HASDAD^TIUSRVLI(DA) I TIUP D D SETDAD^TIUSRVLI(.TIUY,DA,.TIUI**,1**)  . . . N TIUPT  . . . S TIUPT=$P($G(^TIU(8925,+DA,0)),"^",6)  . . . I TIUPT]"",'$D(^TIU(8925,"DAD",TIUPT,DA)) S ^TIU(8925,"DAD",TIUPT,DA)=""  . . . I TIUPT="" S TIUPT=$G(^TIU(8925,+DA,21))  . . . I '$D(^TIU(8925,TIUPT,0)) S $P(@TIUY@(TIUI),"^",16)=1,$P(@TIUY@(TIUI),"^",14)=1 Q  . . I +$$HASKIDS^TIUSRVLI(DA) D |
| **. . . ;ADD IF PREVIOUS RETURN THEN REMOVE ENTRY CODE**  . . . D SETKIDS^TIUSRVLI(.TIUY,DA,.TIUI**,1**)  **I +TIUIEN>0 D RESEQ^TIUSRVLI(.TIUY,.TIUI)**  **I +SHOW>0,+DATTIM>0,+DATTIM'>TIME2,+$G(TIUJ)'<OCCLIM D**  **. S DA=LSTDA ;+$G(@TIUY@(TIUI))**  **. S TIUI=$S(SEQUENCE="D":+$G(TIUI)+1,1:+$G(TIUI)-1)**  **. N TMPDT S TMPDT=9999999-TIME2**  **. S @TIUY@(TIUI)=DA\_"^SHOW MORE^^^^^^^^^^^^"\_$S(+$G(CONTEXT):CONTEXT,1:1)\_"^"**  **. S @TIUY@("INDX",0,TIUI)=""**  Q  ;  ACLAU(TIUY,CLASS,AUTHOR,DFN,TIME1,TIME2,SEQUENCE,INCUND) ; Unsigned  N DATTIM,DA,ROOT,TIUI  S ROOT=$NA(^TIU(8925,"ACLAU",CLASS,AUTHOR,DFN))  S DATTIM=TIME1-.0000001  F S DATTIM=$O(@ROOT@(DATTIM)) Q:DATTIM'>0!(DATTIM>TIME2) D  . S DA=0 F S DA=$O(@ROOT@(DATTIM,DA)) Q:DA'>0 D  . . I +$G(^TIU(8925,+DA,0))'>0 K @ROOT@(DATTIM,DA) Q  . . I +$P($G(^TIU(8925,DA,0)),U,5)>6 K @ROOT@(DATTIM,DA) Q  . . S TIUI=$S(SEQUENCE="D":+$G(TIUI)+1,1:+$G(TIUI)-1)  . . Q:+$D(@TIUY@("INDX",DA))  . . ;Q:'+$$ISDOC^TIUSRVLI(DA) ;If record not a document Quit  . . ; Selectively filter DELETED or RETRACTED records  . . I +$P($G(^TIU(8925,DA,0)),U,5)>13,'+$$CANDO^TIULP(DA,"VIEW",DUZ) Q  . . S @TIUY@(TIUI)=DA\_U\_$$RESOLVE(DA)  . . S @TIUY@("INDX",DA,TIUI)=""  . . Q:+$G(SHOWADD)=0  . . S TIUP=+$$HASDAD^TIUSRVLI(DA) I TIUP D D SETDAD^TIUSRVLI(.TIUY,DA,.TIUI**,0**)  . . . N TIUPT  . . . S TIUPT=$P($G(^TIU(8925,+DA,0)),"^",6)  . . . I TIUPT]"",'$D(^TIU(8925,"DAD",TIUPT,DA)) S ^TIU(8925,"DAD",TIUPT,DA)=""  . . . I TIUPT="" S TIUPT=$G(^TIU(8925,+DA,21))  . . . I '$D(^TIU(8925,TIUPT,0)) S $P(@TIUY@(TIUI),"^",16)=1,$P(@TIUY@(TIUI),"^",14)=2 Q  . . I +$$HASKIDS^TIUSRVLI(DA) D SETKIDS^TIUSRVLI(.TIUY,DA,.TIUI**,0**)  I +$G(INCUND) D GETUND^TIUSRVLI(.TIUY,CLASS,DFN,TIME1,TIME2,.TIUI,SEQUENCE)  Q  ;  ACLEC(TIUY,CLASS,EXCOSIGN,DFN,TIME1,TIME2,SEQUENCE) ; Uncosigned  N DATTIM,DA,ROOT,TIUI  S ROOT=$NA(^TIU(8925,"ACLEC",CLASS,EXCOSIGN,DFN))  S DATTIM=TIME1-.0000001  F S DATTIM=$O(@ROOT@(DATTIM)) Q:DATTIM'>0!(DATTIM>TIME2) D  . S DA=0 F S DA=$O(@ROOT@(DATTIM,DA)) Q:DA'>0 D  . . I +$G(^TIU(8925,+DA,0))'>0 K @ROOT@(DATTIM,DA) Q  . . S TIUI=$S(SEQUENCE="D":+$G(TIUI)+1,1:+$G(TIUI)-1)  . . Q:+$D(@TIUY@("INDX",DA))  . . ;Q:'+$$ISDOC^TIUSRVLI(DA) ;If record not a document Quit  . . ; Selectively filter DELETED or RETRACTED records  . . I +$P($G(^TIU(8925,DA,0)),U,5)>13,'+$$CANDO^TIULP(DA,"VIEW",DUZ) Q  . . S @TIUY@(TIUI)=DA\_U\_$$RESOLVE(DA)  . . S @TIUY@("INDX",DA,TIUI)=""  . . Q:+$G(SHOWADD)=0  . . S TIUP=+$$HASDAD^TIUSRVLI(DA) I TIUP D D SETDAD^TIUSRVLI(.TIUY,DA,.TIUI**,0**)  . . . N TIUPT  . . . S TIUPT=$P($G(^TIU(8925,+DA,0)),"^",6)  . . . I TIUPT]"",'$D(^TIU(8925,"DAD",TIUPT,DA)) S ^TIU(8925,"DAD",TIUPT,DA)=""  . . . I TIUPT="" S TIUPT=$G(^TIU(8925,+DA,21))  . . . I '$D(^TIU(8925,TIUPT,0)) S $P(@TIUY@(TIUI),"^",16)=1,$P(@TIUY@(TIUI),"^",14)=3 Q  . . I +$$HASKIDS^TIUSRVLI(DA) D SETKIDS^TIUSRVLI(.TIUY,DA,.TIUI**,0**) |
| Q  ;  ACLSB(TIUY,CLASS,SIGNEDBY,DFN,TIME1,TIME2,SEQUENCE) ; Signed, by author  N DATTIM,DA,ROOT,TIUI  S ROOT=$NA(^TIU(8925,"ACLSB",CLASS,SIGNEDBY,DFN))  S DATTIM=TIME1-.0000001  F S DATTIM=$O(@ROOT@(DATTIM)) Q:DATTIM'>0!(DATTIM>TIME2) D  . S DA=0 F S DA=$O(@ROOT@(DATTIM,DA)) Q:DA'>0 D  . . I +$G(^TIU(8925,+DA,0))'>0 K @ROOT@(DATTIM,DA) Q  . . S TIUI=$S(SEQUENCE="D":+$G(TIUI)+1,1:+$G(TIUI)-1)  . . Q:+$D(@TIUY@("INDX",DA))  . . ;Q:'+$$ISDOC^TIUSRVLI(DA) ;If record not a document Quit  . . ; Selectively filter DELETED or RETRACTED records  . . I +$P($G(^TIU(8925,DA,0)),U,5)>13,'+$$CANDO^TIULP(DA,"VIEW",DUZ) Q  . . S @TIUY@(TIUI)=DA\_U\_$$RESOLVE(DA)  . . S @TIUY@("INDX",DA,TIUI)=""  . . Q:+$G(SHOWADD)=0  . . I +$$HASDAD^TIUSRVLI(DA) D  . . . S TIUP=+$$HASDAD^TIUSRVLI(DA) I TIUP D D SETDAD^TIUSRVLI(.TIUY,DA,.TIUI**,0**)  . . . N TIUPT  . . . S TIUPT=$P($G(^TIU(8925,+DA,0)),"^",6)  . . . I TIUPT]"",'$D(^TIU(8925,"DAD",TIUPT,DA)) S ^TIU(8925,"DAD",TIUPT,DA)=""  . . . I TIUPT="" S TIUPT=$G(^TIU(8925,+DA,21))  . . . I '$D(^TIU(8925,TIUPT,0)) S $P(@TIUY@(TIUI),"^",16)=1,$P(@TIUY@(TIUI),"^",14)=1 Q  . . I +$$HASKIDS^TIUSRVLI(DA) D SETKIDS^TIUSRVLI(.TIUY,DA,.TIUI**,0**)  Q |

##### PSPO # 934 – Display of non-VA Med Confusion

###### Routine ORCMEDT2

| Routines | Activities | | | |
| --- | --- | --- | --- | --- |
| **Routine Name** | ORCMEDT2 | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RTM** |  | | | |
| **Related Options** |  | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
|  | Called by CPRS GUI via RPC call | HASDAD^TIUSRVLI |

| 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: DFN  Definition: Patient Internal Entry Number | | | | |
| **Output Attribute Name and Definition** | Name: ORY  Definition: Total count of all signed notes, minus addendums and child interdisciplinary notes. | | | | |

| Current Logic |
| --- |
| **Original ADDITM tag:**  ADDITM ; -- Add new item(s) to menu  N DA,DR,DIE,DONE,OREBLD,DLG,ROW,COL,POS  D FULL^VALM1 S VALMBCK="R"  S DA(1)=+ORMENU,DIE="^ORD(101.41,"\_DA(1)\_",10,"  S:'$D(^ORD(101.41,+ORMENU,10,0)) ^(0)="^101.412AI^^"  S (DONE,OREBLD)=0 F D Q:DONE W !  . S DLG=$$ITEM^ORCMEDT0 I DLG'>0 S DONE=1 Q  ADD1 . S ROW=$$ROW Q:ROW="^"  . S COL=$$COL Q:COL="^" S POS=ROW\_"."\_COL  . I $D(^ORD(101.41,+ORMENU,10,"B",POS)) W $C(7),!,"There is another ite  m in this position already!" G:'$$SHIFT("down") ADD1 D INSERT(POS)  . S DA=$$NEXT(POS) Q:'DA S OREBLD=1  . S DR="2////"\_DLG\_";4;3" D ^DIE  I OREBLD W !!,"Rebuilding menu display ..." D INIT^ORCMEDIT S VALMBCK="  R"  Q  **Original EDIT tag:**  EDIT ; -- Edit item(s) in menu  N STOP,OREBLD,DA,DR,DIE,OR0,ORY,ORI,PTR,ROW,COL,POS,NEWPOS,ORDG,P  D FULL^VALM1,SELECT("change",.ORY) S VALMBCK="R" Q:ORY'>0  S (STOP,OREBLD,ORI)=0 F S ORI=$O(ORY(ORI)) Q:ORI'>0 D Q:STOP W !  . S DA=+ORY(ORI),OR0=$G(^ORD(101.41,+ORMENU,10,DA,0)),PTR=+$P(OR0,U,2)  . S DIE("NO^")="OUTOK",DA(1)=+ORMENU,DIE="^ORD(101.41,"\_DA(1)\_",10,"  . I PTR S P=$$ITEM^ORCMEDT0(PTR) S:'P STOP=1 I P S DR=$S(P'=PTR:"2////"  \_P\_";",1:"")\_"4;3" D ^DIE S:$D(Y) STOP=1 ;^ or timeout  . I 'PTR D Q:STOP ;edit #4&5  . . N X1,X2 S X1=$P(OR0,U,4),X2=$P(OR0,U,5)  ED0 . . S X1=$$TEXT($P(OR0,U,4)) I X1="^" S STOP=1 Q  . . I X1="@" G:'$$SURE ED0 S DIK=DIE,(STOP,OREBLD)=1 D ^DIK Q  . . S X2=$$OUTPUT(X2) S:X2="^" STOP=1,X2=$P(OR0,U,5)  . . S ^ORD(101.41,+ORMENU,99)=$H,$P(^(10,DA,0),U,4,5)=X1\_U\_X2  . S:OR0'=$G(^ORD(101.41,+ORMENU,10,DA,0)) OREBLD=1,OR0=$G(^(0))  . Q:$G(STOP) S PTR=+$P(OR0,U,2)  ED1 . S POS=$P(OR0,U),ROW=$$ROW($P(POS,".")) I ROW="^" S STOP=1 Q  . S COL=$$COL($P(POS,".",2)) I COL="^" S STOP=1 Q  . S NEWPOS=ROW\_"."\_COL G:POS=NEWPOS ED2 ; no change  . I $D(^ORD(101.41,+ORMENU,10,"B",NEWPOS)) W $C(7),!,"There is another  item in this position already!" G:'$$SHIFT("down") ED1 D INSERT(NEWPOS)  . S OREBLD=1,DR=".01///"\_NEWPOS D ^DIE  ED2 . Q:'PTR S TYPE=$P($G(^ORD(101.41,PTR,0)),U,4),ORDG=+$P($G(^(0)),U,5)  . I TYPE'="M",$$EDTITM(TYPE) D QCK0^ORCMEDT1(PTR):TYPE="Q",SET0^ORCMEDT  1(PTR):TYPE="O",EN1^ORCMEDT3(PTR):TYPE="D"  I OREBLD W !!,"Rebuilding menu display ..." D INIT^ORCMEDIT S VALMBCK="  R"  Q |

| Modified Logic (Changes are in bold) |
| --- |
| **Original ADDITM tag:**  ADDITM ; -- Add new item(s) to menu  N DA,DR,DIE,DONE,OREBLD,DLG,ROW,COL,POS**,DSPTXT**  D FULL^VALM1 S VALMBCK="R"  S DA(1)=+ORMENU,DIE="^ORD(101.41,"\_DA(1)\_",10,"  S:'$D(^ORD(101.41,+ORMENU,10,0)) ^(0)="^101.412AI^^"  S (DONE,OREBLD)=0 F D Q:DONE W !  . S DLG=$$ITEM^ORCMEDT0 I DLG'>0 S DONE=1 Q  **. S DSPTXT=$P($G(^ORD(101.41,DLG,0)),U,2)**  ADD1 . S ROW=$$ROW Q:ROW="^"  . S COL=$$COL Q:COL="^" S POS=ROW\_"."\_COL  . I $D(^ORD(101.41,+ORMENU,10,"B",POS)) W $C(7),!,"There is another ite  m in this position already!" G:'$$SHIFT("down") ADD1 D INSERT(POS)  . S DA=$$NEXT(POS) Q:'DA S OREBLD=1  . S DR="2////"\_DLG**\_$S(DSPTXT="":";4",1:";4//"\_DSPTXT)\_"**;3" D ^DIE  I OREBLD W !!,"Rebuilding menu display ..." D INIT^ORCMEDIT S VALMBCK="  R"  Q  **Original EDIT tag:**  EDIT ; -- Edit item(s) in menu  N STOP,OREBLD,DA,DR,DIE,OR0,ORY,ORI,PTR,ROW,COL,POS,NEWPOS,ORDG,P**,DSPTX**  **T**  D FULL^VALM1,SELECT("change",.ORY) S VALMBCK="R" Q:ORY'>0  S (STOP,OREBLD,ORI)=0 F S ORI=$O(ORY(ORI)) Q:ORI'>0 D Q:STOP W !  . S DA=+ORY(ORI),OR0=$G(^ORD(101.41,+ORMENU,10,DA,0)),PTR=+$P(OR0,U,2)  . S DIE("NO^")="OUTOK",DA(1)=+ORMENU,DIE="^ORD(101.41,"\_DA(1)\_",10,"  . I PTR S P=$$ITEM^ORCMEDT0(PTR) S:'P STOP=1 I P **D**  **.. S DSPTXT=$P($G(^ORD(101.41,P,0)),U,2)**  **.. S DR=$S(P'=PTR:"2////"\_P\_";",1:"")\_$S(DSPTXT="":"4",1:"4//"\_DSPTXT)\_**  **";3"**  **.. D ^DIE S:$D(Y) STOP=1 ;^ or timeout**  . I 'PTR D Q:STOP ;edit #4&5  . . N X1,X2 S X1=$P(OR0,U,4),X2=$P(OR0,U,5)  ED0 . . S X1=$$TEXT($P(OR0,U,4)) I X1="^" S STOP=1 Q  . . I X1="@" G:'$$SURE ED0 S DIK=DIE,(STOP,OREBLD)=1 D ^DIK Q  . . S X2=$$OUTPUT(X2) S:X2="^" STOP=1,X2=$P(OR0,U,5)  . . S ^ORD(101.41,+ORMENU,99)=$H,$P(^(10,DA,0),U,4,5)=X1\_U\_X2  . S:OR0'=$G(^ORD(101.41,+ORMENU,10,DA,0)) OREBLD=1,OR0=$G(^(0))  . Q:$G(STOP) S PTR=+$P(OR0,U,2)  ED1 . S POS=$P(OR0,U),ROW=$$ROW($P(POS,".")) I ROW="^" S STOP=1 Q  . S COL=$$COL($P(POS,".",2)) I COL="^" S STOP=1 Q  . S NEWPOS=ROW\_"."\_COL G:POS=NEWPOS ED2 ; no change  . I $D(^ORD(101.41,+ORMENU,10,"B",NEWPOS)) W $C(7),!,"There is another  item in this position already!" G:'$$SHIFT("down") ED1 D INSERT(NEWPOS)  . S OREBLD=1,DR=".01///"\_NEWPOS D ^DIE  ED2 . Q:'PTR S TYPE=$P($G(^ORD(101.41,PTR,0)),U,4),ORDG=+$P($G(^(0)),U,5)  . I TYPE'="M",$$EDTITM(TYPE) D QCK0^ORCMEDT1(PTR):TYPE="Q",SET0^ORCMEDT  1(PTR):TYPE="O",EN1^ORCMEDT3(PTR):TYPE="D"  I OREBLD W !!,"Rebuilding menu display ..." D INIT^ORCMEDIT S VALMBCK="  R"  Q |

##### Templates

Table 17: Templates

| Templates | Description | | | |
| --- | --- | --- | --- | --- |
| **Template Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **RSD** |  | | | |
| **Template Type** | Sort | Input | Print | Other |
| **Related Options** |  | | | |

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

| Routines | Description |
| --- | --- |
| **Data Dictionary (DD) References** |  |
| **Global References** |  |

##### Bulletins

Table 19: Bulletins

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

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

| Routines | Description |
| --- | --- |
| **Mail Subject** |  |
| **Mail Group** |  |
| **Parameters** |  |
| **Data Dictionary (DD) References** |  |

##### Data Entries Affected by the Design

SIGN/SYMPTOMS (#120.83)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| NATIONAL SIGN/SYMPTOM (#1) |  | 1:NATIONAL SIGN/SYMPTOM |
| SYNONYM (#2) |  | UNK |
| EFFECTIVE DATE/TIME (#99.991) |  | TO BE DETERMINED |

**D/C Order by Adverse Reaction Request #20080226**

ORDER REASON (#100.03)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| SYNONYM (#.03) |  | ADR |
| INACTIVE(#.04) |  | 0:ACTIVE |
| PACKAGE (#.05) |  | ORDER ENTRY/RESULTS REPORTING |
| NATURE OF ACTIVITY (#.07) |  | REJECTED ?? |

###### CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4)

TIU PERSONAL PREFERENCES (#8926)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| USE TEMPLATE REQD FLD HLITE (#.21) |  | 0: OFF  1: ON |
| TEMPLATE REQD FLD HLITE COLOR (#.22) |  | NUMBER REPRESENTING GUI HILITE COLOR |
| TEMPLATE REQD FLD NAV LOCATION (#.23) |  | 0: TOP  1: LEFT  2: RIGHT  3: BOTTOM |

###### NSR #20110210 Create Separate Alert for Prosthetics Requests Screening Call Record

OE/RR NOTIFICATIONS (#100.9)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| NUMBER |  | 89 |
| NAME (#.01) |  | PROSTHETICS CONSULT UPDATED |
| PACKAGE ID (#.02) |  | OR |
| MESSAGE TEXT (#.03) |  | Prosthetics Consult Updated |
| MESSAGE TYPE (#.04) |  | PACKAGE PROVIDES A VARIABLE MESSAGE |
| ENTRY POINT (#.05) |  | UPCON |
| ACTION FLAG (#.06) |  | RUN ROUTINE |
| ROUTINE NAME (#.07) |  | ORB3FUP2 |
| RELATED PACKAGE (#1.5) |  | OR |
| DESCRIPTION (#4) |  | Triggered within Consult/Request package when a comment is added to  a prosthetics consult or the prosthetics consult is scheduled. Recipients are determined by the Consult/Request package. It is deleted upon completion of the  follow-up action. |

###### NSR #20090509 (15) – Park a Prescription

PRESCRIPTION FILE (#52)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| MAIL/WINDOW/PARK (#11) | **MAIL/WINDOW**  'W' FOR WINDOW;  'M' FOR MAIL; | **MAIL/WINDOW/PARK**  'W' FOR WINDOW;  'M' FOR MAIL;  'P' FOR PARK; |
| MAIL/WINDOW/PARK (52.1,2) | **MAIL/WINDOW**  'W' FOR WINDOW;  'M' FOR MAIL; | **MAIL/WINDOW/PARK**  'W' FOR WINDOW;  'M' FOR MAIL;  'P' FOR PARK; |

OUTPATIENT SITE FILE (#59)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| PARK FUNCTION? (#2009) |  | '0' FOR NO;  '1' FOR YES; |

PENDING OUTPATIENT ORDERS FILE (#52.41)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| PICKUP ROUTING (#19) | Mail/Window/Administered in Clinic  'M' FOR MAIL;  'C' FOR ADMINISTERED IN CLINIC  'W' FOR WINDOW; | Mail/Window/Administered in Clinic **/Parked**  'M' FOR MAIL;  'C' FOR ADMINISTERED IN CLINIC  'W' FOR WINDOW;  **'P' FOR PARK;** |

###### NSR # 20071211 (23) – Changes to Allergy/Pharmacy Packages

ORDER CHECK INSTANCES FILE (#100.05)

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| COMMENT ON REMOTE ALLERGY (#11) – one of the DRUG ALLERGIES (#17) SUB-FIELDS |  | A pre-defined or free text comment in response to remote allergies returned during the Order Check processing of a medication order. |

ORDER CHECK OVERRIDE REASONS (#100.04 -- ^ORD(100.04))

Table 20: Data Entries Affected by the Design

| Field Name | Current Value | New Value |
| --- | --- | --- |
| NAME (#.01) | New file | Benefit of Therapy Outweighs Risk  Patient tolerating current therapy with this medication  Previous Adverse Reaction signs/symptoms managed by patient  Renewal of Current Therapy  Will Monitor Closely for Adverse Effects  Documentation of Allergy/Adverse Reaction is in Error  Documentation of Allergy/Adverse Reaction is to different agent in same drug class |
| SYNONYM (#.02) | New file | BEN  PAT  PRE  REN  WILL  DOA  DOCAA |
| TYPE(#.03) | New file | 'A' FOR ALLERGY;  'O' FOR ORDER CHECK;  'B' FOR BOTH;  'R' FOR REMOTE COMMENT; |
| ACTIVE(#.04) | New file | '0' FOR NO;  '1' FOR YES; |

###### NSR # 20100101 (26) – Indications for Use for Prescription and Medication Orders

###### Pharmacy Orderable Items (#50.7)

**Table 20: Data Entries Affected by the Design**

| Field Name | Current Value | New Value |
| --- | --- | --- |
| MOST COMMON INDICATION FOR USE (#14) |  | Most Common Indication is used as the default indication for a medication |
| INDICATION FOR USE (#13) |  | Multiple field to contain indications for medications |

##### Unique Record(s)

SIGN/SYMPTOMS (#120.83)

Table 21: Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
| NAME (#.01) |  | UNKNOWN |

###### D/C Order by Adverse Reaction Request #20080226

ORDER REASON (#100.03)

Table 21: Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
| NAME (#.01) | N/A | Allergy/Adverse Reaction |

###### Add Address of Performing Lab to Reports in CPRS Health Summary (#20081206)

Table 21: Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
| NUMBER (#.001) |  | 1608 |
| NAME (#.01) |  | ORRPW PHARMACY ALLERGY |

###### CWAD Post Auto-Demotion Rules Request (#20080307) (RSD #2.6.4)

TIU PERSONAL PREFERENCES (#8926)

Table 21: Unique Record ID

| Field Name(s) | Current Value | New Value |
| --- | --- | --- |
| USER NAME (#.01) |  | UNKNOWN |

##### File or Global Size Changes

Table 22: File or Global Size Changes

| File/Global Name(s) | Estimated Increase | Estimated Decrease |
| --- | --- | --- |
|  |  |  |

##### Mail Groups

Table 24: Mail Groups

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

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

| Mail Groups | Instructions | |
| --- | --- | --- |
| **Data Dictionary (DD) References** |  | |
| **Related Protocols** |  | |
| **Mail Group Description** |  | |
| **Self-Enrollment Allowed** | Yes | No |
| **Type** | Public | Private |

##### Security Keys

Table 26: Security Keys

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

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

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

| Current Logic |
| --- |
|  |

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

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

##### Options

Table28: Options

| Options | Activities | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Option Name** | CWAD/Postings Auto-Demotion Setup [TIU CWAD AUTO-DEMOTION] | | | | | | | | | | |
| **Enhancement Category** | New | Modify | | | | Delete | | | No Change | | |
| **Associated Menu Options that will invoke this reference** | Text Integration Utilities (MIS Manager)[ TIU MAIN MENU MGR] | | | | | | | | | | |
| **Data Passing** | Input | | Output | | Both | | | Global Reference | | | Local Reference |
| **Menu Text Description** | This option is used to setup CWAD Auto-Demotion. | | | | | | | | | | |
| **Option Type** | Edit | | | Print | | | Menu | | | Inquire | |
| Action | | | Run Routine | | | Other | | |  | |
| **Associated Routine** | EN^TIUCWDUT | | | | | | | | | | |
| **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

Table 30: Protocols

| Protocols | Activities | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Protocol Name** |  | | | | | | | | | | |
| **Enhancement Category** | New | | Modify | | | | Delete | | No Change | | |
| **Associated Protocols** |  | | | | | | | | | | |
| **Data Passing** | Input | Output | | | Both | | | Global Reference | | | Local Reference |
| **Item Text Description** | N/A | | | | | | | | | | |
| **Protocol Type** | Action | | | Menu | | Protocol | | | | Protocol Menu | |
| Limited Protocol | | | | | Extended Action | | | | Dialog | |
| Other | | | | | | | | | | |
| **Associated Routine** |  | | | | | | | | | | |

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

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

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

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

##### Remote Procedure Call (RPC)

Table 32: RPCs

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWDX2 DCREASON' | | |
| **TAG^RTN** |  | | |
| **Input Parameters** |  | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20080226 - | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWU NEWPERS' | | |
| **TAG^RTN** |  | | |
| **Input Parameters** | [DFN,1,'',True] | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | **RSD #2.6.11 NSR 20110606 (Similar Provider Names)** | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | ORWTIU TEMPLATE PREFERENCES | | |
| **TAG^RTN** | REQDFLD^ORWTIU | | |
| **Input Parameters** | [DFN,1,'',True] | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | **RSD #2.6.12 NSR# 20100706 (Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields)** | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWDXC ALLERGY' | | |
| **TAG^RTN** | ALLERGY^ORWDXC | | |
| **Input Parameters** | DFN – Patient identifier, FID – Literal ‘PSI’, ORL , OIL – Medication id | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20071211 - **Changes to Allergy Pharmacy Package** | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWDXC REASON' | | |
| **TAG^RTN** | REASON^ORWDXC | | |
| **Input Parameters** | TYP – Reason Type, DFN – Patient identifier, OID – Medication id | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20071211 - **Changes to Allergy Pharmacy Package** | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWORB PROUSER' | | |
| **TAG^RTN** | PROUSER^ORWORB | | |
| **Input Parameters** | StartDate, EndDate | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR20081008 - CPRS Notification Alert Processing | | |

| RPCs | Activities | | |
| --- | --- | --- | --- |
| **Name** | 'ORWTPR GETARCHP' | | |
| **TAG^RTN** | GETARCHP^ORWTPR | | |
| **Input Parameters** | None | | |
| **Results Array** | Single Value | Array | Word Processing |
| Global Array | Global Instance |  |
| **Description** | NSR 20081008 (**Alert Notification – Alert Processing**) **Returns a value based on the ORB ARCHIVE PERIOD parameter** | | |

##### Constants Defined in Interface

Table 33: Constants Defined in Interface

| Name | Description |
| --- | --- |
|  |  |

##### Variables Defined in Interface

Table 34: Variables Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
| DupDFN | **Int64** | **SD #2.6.11 NSR 20110606 (Similar Provider Names)** |

##### Types Defined in Interface

Table 35: Types Defined in Interface

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### GUI

Table 36: GUI

| Unit Name | Description |
| --- | --- |
| frmDCOrders | NSR20080226 - The GUI form was modified to add additionally, a check to the "Discontinue/Cancel Orders screen" selection list to validate the selection when " Allergy/Drug Adverse Reaction" is chosen. |

| Unit Name | Description |
| --- | --- |
| fOrders.pas | Option 1. Has an array list of items passed in to the main function call for this procedure.  Option 2. The function now is only passed an individual array item in a loop rather than passing the entire array, thus allowing for individual processing of Critical High Orders |

| Unit Name | Description |
| --- | --- |
| frmOCSession.pas | **RSD #2.6.18 NSR 20101203 Critical-High Order Check Display.**  The purpose of this NSR is to allow for assigning individual override reasons and comments to each order rather than just an overall reason given for all orders as it currently stands in CPRS. |
| fNotes.pas | **RSD #2.6.25 NSR 20070817 Progress Notes Display Misleading**  Function LoadNotes was modified so that under a certain condition selected objects/variables were not cleared;  Procedure tvNotesChange was modified so that new records could be retrieved and the notes properly positioned in the tree as well as removing information that was duplicated with each retrieval to give a continuous list of selected records.  The totals for total number of records for a person and the total number displayed were accurate. |

|  |  |
| --- | --- |
| uDocTree.pas | **RSD #2.6.25 NSR 20070817 Progress Notes Display Misleading**  Several new variables were set to values in this unit in conjunction with the overall purpose of the NSR. |

| Unit Name | Description |
| --- | --- |
| fConsults.pas | NSR20110210 – Added follow-up action to delete notification after it has been processed. |
| fFrame.pas | NSR20110210 – Added follow-up action to move user to Consults Tab upon processing the prosthetics notification. |
| uConst.pas | NSR20110210 – Added new constant value for the prosthetics notification. |

###### Modify the Similar Patients form to also display similar providers and cosigners (NSR#20110606)

Summary of changes

1. The RPCs “ORWU2 COSIGNER”, “ORWU NEWPERS” were updated to support search for “similar” names for validation of Provider and Cosigner fields.
2. Class TORComboBox was modified to add validation of the selected name with the updated RPC.
3. Class TfrmDupPts was modified to make it possible to use the class form for selection of single Provider or Cosigner (in addition to selection of a single patient). The class form was provided with custom constructor and the unit routines were updated to support the exception lists.
4. Classes that supported selection of the Provider or Cosigner (units fEncnt, fEditConsult, fNoteProps, fDCSummProps, fNoteCPFields, fOrdersAlert, fODCounsult, fODProc) were modified with validation of the names of the selected persons based on updated TORComboBox.
5. Classes that were used for selection of multiple names from the list (units fAddlSigners, fAlertForward, fVisitType) were modified to use common validation routine.
6. No changes were made to the form appearance.

Files changed

| Unit Name | Summary of changes |
| --- | --- |
| fAddlSigners | Updated to support validation of provider names. Cosigner name is not validated as the “Cosigner” field is not editable |
| fAlertForward | Updated to support validation of Provider name. |
| fDCSummProps | Updated to support validation of Author and Cosigner names. Processing of clicks on cmdOK, cmdCancel buttons changed. |
| fDupPts | Updated to generalize the selection of items from the list |
| fEncnt | Updated to support validation of names of Author (Dictator) and Attending Physician fields. The enhanced version of TORComboBox component used. The form updated to use OnCloseQuery instead of custom flag |
| fNoteCPFieds | Updated to support validation of names of Author field. The enhanced version of TORComboBox component used. |
| fNoteProps | Updated to support validation of names of Author and Expected Cosigner fields. The enhanced version of TORComboBox component used. |
| fPtSel | Updated to correlate changes to the fDupPts unit with validation of the patient name. |
| fPtSelOptns | Updated to support validation of names of Provider List field. |
| rCore | Added functions providing results of names validation on Server. |
| fEditConsult | Updated to support validation of names of Attention field. The enhanced version of TORComboBox component used. |
| fODProc | Updated to support validation of names of Attention field. The enhanced version of TORComboBox component used. |
| fODConsult | Updated to support validation of names of Attention field. The enhanced version of TORComboBox component used. |
| fVisitType | Updated to support validation of Available providers names. |
| fOrdersAlert | Updated to support validation of names of Alert Recipient field. The enhanced version of TORComboBox component used. The form updated to use OnCloseQuery instead of custom flag |
| uSimilarNames | New unit implementing changes to TORComboBox required for validation of the selected name. |
| CPRSChart.dpr | Updated to include reference on unit uSimilarNames |

Unit fAddlSigners

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| procedure TfrmAddlSigners.btnAddSignersClick(Sender: TObject); | Modified | Code modified to add validation:  procedure TfrmAddlSigners.btnAddSignersClick(Sender: TObject);  var  i: integer;  DUZ: Int64;  begin  if cboSrcList.ItemIndex = -1 then  exit;  if UserInactive(cboSrcList.ItemID) then  if (InfoBox(fNoteProps.TX\_USER\_INACTIVE, TC\_INACTIVE\_USER, MB\_OKCANCEL)= IDCANCEL) then  exit;  if (DstList.SelectByID(cboSrcList.ItemID) <> -1) then  begin  InfoBox(TX\_DUP\_SIGNER, TX\_SIGNER\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  for i := 0 to FExclusions.Count-1 do  if (Piece(FExclusions.Strings[i],U,1) = cboSrcList.ItemID) then  begin  InfoBox(TX\_BAD\_SIGNER, TX\_SIGNER\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  // verify if filtering by PROVIDER key is needed!  DUZ := getProviderIdCheckedForSimilarName(cboSrcList.ItemID,'PROVIDER',DstList.Items);  if DUZ > 0 then  begin  cboSrcList.SelectByIEN(DUZ);  DstList.Items.Add(cboSrcList.Items[cboSrcList.Itemindex]) ;  end;  btnRemoveSigners.Enabled := DstList.SelCount > 0;  btnRemoveAllSigners.Enabled := DstList.Items.Count > 0;  end; |
| function TfrmAddlSigners.CosignerOK: Boolean; | Modified | Code modified to add validation of the selected name  function TfrmAddlSigners.CosignerOK: Boolean;  var  sError: String;  begin  Result := False;  if not cboCosigner.Enabled then  begin  Result := True;  Exit;  end;  if cboCosigner.ItemIndex < 0 then  begin  InfoBox(TX\_NO\_DELETE, TX\_NO\_DELETE\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  if cboCosigner.NeedsValidation then  begin // check if there are cosigners with similar names  sError := cboCosigner.ValidationReport;  if sError <> '' then  begin  InfoBox(sError, TX\_NO\_COSIGNER\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  end;  case FTabID of  CT\_NOTES, CT\_CONSULTS:  if (not CanCosign(TitleForNote(FNoteIEN), 0, cboCosigner.ItemIEN, FRefDate)) then  begin  InfoBox(cboCosigner.Text + TX\_NO\_COSIGNER, TX\_NO\_COSIGNER\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  CT\_DCSUMM:  if not IsUserAProvider(cboCosigner.ItemIEN, FMNow) then  begin  InfoBox(cboCosigner.Text + TX\_NO\_COSIGNER, TX\_NO\_COSIGNER\_CAP, MB\_OK or MB\_ICONWARNING);  Exit;  end;  end;  Result := True;  end; |

Unit fAlertForward

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Procedure  TfrmAlertForward.cboSrcListMouseClick(Sender: TObject); |  | procedure TfrmAlertForward.cboSrcListMouseClick(Sender: TObject);  var  DUZ: Int64;  begin  // if cboSrcList.ItemIndex = -1 then exit;  // if (DstList.SelectByID(cboSrcList.ItemID) <> -1) then  // InfoBox(TX\_DUP\_RECIP, TX\_RECIP\_CAP, MB\_OK or MB\_ICONWARNING)  // else  if cboSrcList.ItemIndex > -1 then  begin  DUZ := getProviderIdCheckedForSimilarName(cboSrcList.ItemID,'PROVIDER',DstList.Items);  if DUZ > 0 then  begin  cboSrcList.SelectByIEN(DUZ);  DstList.Items.Add(cboSrcList.Items[cboSrcList.Itemindex]);  if ScreenReaderSystemActive then  GetScreenReader.Speak(Piece(cboSrcList.Items[cboSrcList.Itemindex],U,2) +  ' Added to ' + DstLabel.Caption);  btnRemoveAlertFwrd.Enabled := DstList.SelCount > 0;  btnRemoveAllAlertFwrd.Enabled := DstList.Items.Count > 0;  end;  end;  end; |
|  |  |  |

Unit fDCSummProps

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Definition of TORComboBox | New | Type definition added to replace standard TORComboBox with the one defined in uSimilarNames unit  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Form methods TfrmDCSummProperties = class(TfrmBase508Form) | New | New methods added to the form  procedure cmdCancelMouseDown(Sender: TObject; Button: TMouseButton;  Shift: TShiftState; X, Y: Integer);  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);  procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);  procedure FormCreate(Sender: TObject);  procedure calSummChange(Sender: TObject); |
| Form methods TfrmDCSummProperties = class(TfrmBase508Form) | Removed | Methods removed as the form was updated to use the standard OnFormCloseQuery instead of custom flag  procedure cmdOKClick(Sender: TObject);  procedure cmdCancelClick(Sender: TObject);  procedure cboAttendingExit(Sender: TObject);  procedure cboAuthorExit(Sender: TObject); |
| Form private method TfrmDCSummProperties = class(TfrmBase508Form) | New | New function added to use for validation of the form fields  function isValidInput:Boolean; |
| Form function  ExecuteDCSummProperties(var ASumm: TEditDCSummRec;  var ListBoxItem: string; ShowAdmissions, IDNoteTitlesOnly: Boolean): Boolean; | Modified | Function was modified to provide the updated components with required data  cboAuthor.ItemName := Copy(lblAuthor.Caption,1,Length(lblAuthor.Caption)-1);  cboAttending.ItemName := Copy(lblCosigner.Caption,1,Length(lblCosigner.Caption)-1);  Result := ShowModal = idOK; // display the form  if Result then |
| Form Procedure  TfrmDCSummProperties.cboNewTitleMouseClick(Sender: TObject); | Modified | Procedure was modified to track changes to the title of the note  procedure TfrmDCSummProperties.cboNewTitleMouseClick(Sender: TObject);  begin  with cboNewTitle do  if (ItemIEN > 0) and (ItemIEN = FLastTitle) then  Exit  else if ItemIEN = 0 then  begin  if FLastTitle > 0 then  SelectByIEN(FLastTitle)  else  ItemIndex := -1;  Exit;  end;  SetCosignerRequired;  if FShowAdmissions and (not pnlAdmission.Visible) then  begin  pnlAdmission.Visible := True;  pnlAdmission.Top := cmdCancel.Top + cmdCancel.Height + 8;  pnlAdmission.Height := Height - pnlAdmission.Top;  end;  FLastTitle := cboNewTitle.ItemIEN;  cboAttending.TitleIEN := FLastTitle;  end; |
| Form procedure  procedure TfrmDCSummProperties.calSummChange(Sender: TObject); | New | Procedure modified to track changes to the date of the note  procedure TfrmDCSummProperties.calSummChange(Sender: TObject);  begin  inherited;  cboAttending.ValidationDate := CalSumm.FMDateTime;  end; |
| Form Procedure  procedure TfrmDCSummProperties.cboAuthorExit(Sender: TObject); | Removed | Procedure removed as OnExit event is not used for validation of the Author field |
| Form Procedure  procedure TfrmDCSummProperties.cboAttendingExit(Sender: TObject); | Removed | Procedure removed as OnExit event is not used for validation of the Attending field |
| Form procedure  procedure TfrmDCSummProperties.cmdOKClick(Sender: TObject); | Removed | Procedure removed. Replaced with function  function TfrmDCSummProperties.isValidInput:Boolean; |
| Function  function TfrmDCSummProperties.isValidInput:Boolean; | New | Function replaces the original version of validation method CMDOK1Click. ValidationReport method of the updated ORComboBox object indicates if the field is populated correctly.  function TfrmDCSummProperties.isValidInput:Boolean;  var  ErrMsg, ItemText, WhyNot: string;  begin  SetCosignerRequired;  ErrMsg := '';  if cboNewTitle.ItemIEN = 0 then  ErrMsg := ErrMsg + TX\_REQ\_TITLE  else if FIDNoteTitlesOnly and  (not CanTitleBeIDChild(cboNewTitle.ItemIEN, WhyNot)) then  ErrMsg := ErrMsg + CRLF + WhyNot;  errMsg := errMsg + cboAuthor.ValidationReport;  if not calSumm.IsValid then  ErrMsg := ErrMsg + TX\_REQ\_REFDATE;  if calSumm.IsValid and (calSumm.FMDateTime > FMNow) then  ErrMsg := ErrMsg + TX\_NO\_FUTURE;  if cboAttending.Visible and (cboAttending.ItemIEN = 0) then  ErrMsg := ErrMsg + TX\_REQ\_COSIGNER;  // if cboAttending.ItemIEN = User.DUZ then ErrMsg := TX\_COS\_SELF;  if (cboAttending.ItemIEN > 0) then  begin  if ((not IsUserAUSRProvider(cboAttending.ItemIEN, FMNow)) or  (not CanCosign(cboNewTitle.ItemIEN, FDocType, cboAttending.ItemIEN,  calSumm.FMDateTime)))  then  ErrMsg := cboAttending.Text + TX\_COS\_AUTH;  ErrMsg := ErrMsg + cboAttending.ValidationReport;  end;  if pnlAdmission.Visible then  with lstAdmissions do  begin  if not Assigned(Selected) then  ErrMsg := TX\_NO\_ADMISSION  else  begin  ItemText:= Strings[Selected.Index];  if (Piece(ItemText, U, 7) = '1') then  begin  FVisitStr := Piece(ItemText, U, 2) + ';' +  Piece(ItemText, U, 1) + ';H';  if (OneNotePerVisit(cboNewTitle.ItemIEN, Patient.DFN, FVisitStr))  then  begin  FEditIEN := 0;  InfoBox(TX\_NO\_MORE\_SUMMS, TC\_NO\_EDIT, MB\_OK);  Selected := nil;  end;  end  else  begin  FAdmitDateTime := Piece(ItemText, U, 1);  FLocation := StrToIntDef(Piece(ItemText, U, 2), 0);  if (MakeFMDateTime(FAdmitDateTime) = -1) or (FLocation = 0) then  ErrMsg := TX\_BAD\_ADMISSION  else  FLocationName := ExternalName(FLocation, 44);  end;  end;  end;  Result := Length(ErrMsg) = 0;  ErrMsg := Trim(ErrMsg);  ShowMsgOn(not Result, ErrMsg, TC\_REQ\_FIELDS);  end; |
| Procedure  procedure TfrmDCSummProperties.cmdCancelMouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer); | Removed | Procedure was removed – replaced with the use of ModalResult property assigned to the button |
| Procedure | New | Procedure allows to close the form by clicking of the button with mouse  procedure TfrmDCSummProperties.cmdCancelMouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  begin  inherited;  ModalResult := mrCancel;  end; |
| Procedure  procedure TfrmDCSummProperties.FormCloseQuery(Sender: TObject;  var CanClose: Boolean); | New | Procedure Validating values entered in the form fields  procedure TfrmDCSummProperties.FormCloseQuery(Sender: TObject;  var CanClose: Boolean);  begin  inherited;  if ModalResult = mrOK then  CanClose := isValidInput;  end; |
| Procedure  procedure TfrmDCSummProperties.FormCreate(Sender: TObject); | New | Procedure sets up the updated TORComboBox components with required data  procedure TfrmDCSummProperties.FormCreate(Sender: TObject);  begin  inherited;  setupSimilarNameSelector(cboAuthor,aPr,'');  cboAuthor.msgNoSelection := TX\_REQ\_AUTHOR;  setupSimilarNameSelector(cboAttending,aCo,'');  cboAttending.msgNoSelection := TX\_REQ\_COSIGNER;  end; |
| Procedure  procedure TfrmDCSummProperties.FormKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState); | New | Procedure allows to close form by clicling Esc key  procedure TfrmDCSummProperties.FormKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if Key = vk\_Escape then  ModalResult := mrCancel;  end; |

Unit fDupPts

The class TfrmDupPts was originally design to let user select one of the patients from the list. The class was updated to generalize selection: the custom constructor of the class was created. The constructor accepts the list of items for selection. With the updated constructor the form was able to serve selection of Providers and Cosigners (in addition to selection of the patients).

The form private property fExceptions is used to track items that should not be selected.

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Private field  fSelector:TSelector | New | Type of the selector. 3 types defined in unit uSimilarnames are supported: Patient, Provider, Cosigner |
| Private field  fItemName:String | New | Field is used for generation of label captions based on the selector type. |
| Private field  fExceptions: TStrings | New | Field keeps list of items that are not allowed for selection |
| Constructor  Constructor CreateSelector(ASelector:TSelector; aList:TStrings; anExceptions:TStrings = nil); | New | Custom constructor to create the form and setup the form properties  constructor TfrmDupPts.CreateSelector(ASelector:TSelector; aList:TStrings; anExceptions:TStrings = nil);  const  fmtCaptionWindow = 'Similar %ss';  fmtCaptionList = ' Please select the correct %s:';  begin  inherited Create(Application);  fSelector := ASelector;  fExceptions := anExceptions;  case fSelector of  APr : fItemName := 'provider';  ACo : fItemName := 'cosigner';  APt : fItemName := 'patient';  end;  pnlHeader.Caption := Format(fmtCaptionList,[fItemName]);  fItemName := uppercase(copy(fItemName,1,1))+copy(fItemName,2,Length(fItemName));  Caption := Format(fmtCaptionWindow,[fItemName]);  if fSelector <> APt then  begin  lboSelPt.Columns[1].Caption := 'Position';  lboSelPt.Columns[1].Width := lboSelPt.Width - lboSelPt.Columns[0].Width - 24;  lboSelPt.Columns.Delete(2);  end;  if Assigned(aList) then  FastAssign(aList, lboSelPt.ItemsStrings);  ResizeAnchoredFormToFont(self);  end; |
| Procedure  procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState); | New | Procedure Alloows to close the form on ESC click.  procedure TfrmDupPts.FormKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if (key = VK\_ESCAPE) then  ModalResult := mrCancel;  end; |
| Procedure  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean); | New | Procedure validates form prior to closing  procedure TfrmDupPts.FormCloseQuery(Sender: TObject; var CanClose: Boolean);  begin  inherited;  if ModalResult <> mrCancel then  begin  CanClose := Length(lboSelPt.ItemID) > 0;  if not CanClose then  InfoBox(' A ' + fItemName + ' has not been selected', 'No '+fItemName + ' Selected', MB\_OK or MB\_ICONWARNING);  if CanClose then  CanClose := ValidateExceptions;  end;  end; |
| Procedure  procedure lboSelPtCustomDrawItem(Sender: TCustomListView; Item: TListItem; | New | Procedure used to provide separate presentation for items that are included in the Exception list  procedure TfrmDupPts.lboSelPtCustomDrawItem(Sender: TCustomListView;  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);  var  sID:String;  begin  inherited;  if assigned(Item) then  begin  sID := lboSelPt.ItemsStrings[Item.Index];  sID := Piece(sID,U,1)+ U;  if assigned(fExceptions) and (pos(sID,fExceptions.Text)> 0) then  lboSelPt.Canvas.Font.Color := clHighlight  else  lboSelPt.Canvas.Font.Color := clWindowText;  end  end; |
| Function  function ValidateExceptions:Boolean; | New | Function validates if the selected item is on fException list  function TfrmDupPts.ValidateExceptions: Boolean;  var  i: Integer;  const  msgRecordAlreadyAdded = 'This Record is already added to the Selection List'+CRLF+  'Please select another Record or Cancel the selection';  begin  Result := True;  if not assigned(fExceptions) then  exit;  for i := 0 to fExceptions.Count -1 do  if (Piece(fExceptions[i],U,1) = lboSelPt.ItemID) then  begin  Result := False;  InfoBox(msgRecordAlreadyAdded,'Duplicate item detected', MB\_OK or MB\_ICONWARNING);  break;  end;  end; |

Unit fEncnt

Form layout was updated to automate positioning and sizing of the components.

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Method  procedure cmdOKClick(Sender: TObject); | Removed | Method replaced with FormCloseQuery |
| Method  procedure FormResize(Sender: TObject); | Removed | Method removed. The re-positioning of the form components is done automatically based on anchor values. |
| Procedure  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean); | New | Procedure validates form prior to closing  procedure TfrmEncounter.FormCloseQuery(Sender: TObject; var CanClose: Boolean);  var  msg: string;  ADate, AMaxDate: TDateTime;  begin  inherited;  if ModalResult = mrCancel then  exit;  if cboPtProvider.ItemID = '' then  begin  InfoBox(msg, TC\_MISSING, MB\_OK);  CanClose := False;  end  else  begin  if (not DoNotNeedLocation) or (DoNotNeedLocation and (FLocation > 0)) then  begin  msg := '';  if FLocation = 0 then msg := TX\_NO\_LOC;  if FDateTime <= 0 then msg := msg + CRLF + TX\_NO\_DATE  else if(pos('.',FloatToStr(FDateTime)) = 0) then msg := msg + CRLF + TX\_NO\_TIME;  msg := msg + cboPtProvider.ValidationReport;  if(msg <> '') then  begin  InfoBox(msg, TC\_MISSING, MB\_OK);  CanClose := False; //Exit;  end  else  begin  ADate := FMDateTimeToDateTime(Trunc(FDateTime));  AMaxDate := FMDateTimeToDateTime(FMToday) + StrToIntDef(FEncFutureLimit, 0);  if ADate > AMaxDate then  if InfoBox(TX\_FUTURE\_WARNING, TC\_FUTURE\_WARNING, MB\_YESNO or MB\_ICONQUESTION) = MRNO then  CanClose := False; //exit;  end;  end;  if FFilter <> NPF\_SUPPRESS then  FProvider := cboPtProvider.ItemIEN;  end;  end; |
| Procedure  procedure cboPtProviderDblClick(Sender: TObject); | New | Procedure performs validation on double click of the selected item  procedure TfrmEncounter.cboPtProviderDblClick(Sender: TObject);  begin  inherited;  if cboPtProvider.NeedsValidation then  begin  if cboPtProvider.isValidSelection(Sender) then  cboPtProvider.NeedsValidation := False;  end  else  ModalResult := mrOK;  end; |
| Procedure  procedure SetLayout; | New | Procedure encapsulates actions required for setting the form properties (replaces original FormShow method)  procedure TfrmEncounter.setLayout;  begin  //CQ7118  if cboPtProvider.Visible then  begin  cmdOK.Left := cboPtProvider.Left + cboPtProvider.Width + 1;  cmdCancel.Left := cboPtProvider.Left + cboPtProvider.Width + 1;  end  else  begin  cmdOK.Left := cmdDateRange.Left;  cmdCancel.Left := cmdDateRange.Left;  end;  cmdCancel.Top := cmdDateRange.Top - cmdCancel.Height - 10;  cmdOK.Top := cmdCancel.Top - cmdOK.Height - 1;  cmdCancel.Top := cmdOK.Top + cmdOK.Height + 1;  // cmdCancel.Width := cmdOK.Width;  //end CQ7118  end; |
| Procedure  procedure UpdateEncounter(PersonFilter: Int64; ADate: TFMDateTime = 0; TIULocation: integer = 0; DelayedOrder: Boolean = False); | Modified | Procedure was modified to implements similar names validation  procedure UpdateEncounter(PersonFilter: Int64; ADate: TFMDateTime = 0; TIULocation: integer = 0; DelayedOrder: Boolean = False);  const  UP\_SHIFT = 85;  var  frmEncounter: TfrmEncounter;  TimedOut: Boolean;  begin  uTIULocation := TIULocation;  if uTIULocation <> 0 then uTIULocationName := ExternalName(uTIULocation, FN\_HOSPITAL\_LOCATION);  frmEncounter := TfrmEncounter.Create(Application);  try  if DelayedOrder = True then frmEncounter.DoNotNeedLocation := True  else frmEncounter.DoNotNeedLocation := False;  TimedOut := False;  ResizeAnchoredFormToFont(frmEncounter);  with frmEncounter do  begin  FFilter := PersonFilter;  FPCDate := ADate;  if PersonFilter = NPF\_SUPPRESS then // not prompting for provider  begin  lblProvider.Visible := False;  cboPtProvider.Visible := False;  lblInstruct.Visible := True;  Caption := TC\_LOCONLY;  Height := frmEncounter.Height - UP\_SHIFT;  end  else // also prompt for provider  begin  // InitLongList must be done AFTER FFilter is set  cboPtProvider.InitLongList(Encounter.ProviderName);  cboPtProvider.SelectByIEN(FProvider);  end;  if ShowModal = mrOK then  begin  // we don't know the original value of DelayReviewChanges but the code was updating it only in some cases  DelayReviewChanges := DelayReviewChanges or  (DEAContext and // I can't identify cases DEAContext is TRUE when UpdateEncounter is called :(  ((Assigned(Changes.Orders)) and (Changes.Count > 0)) and (Encounter.Provider <> FProvider));  if (PersonFilter = NPF\_SUPPRESS) // NPF\_SUPPRESS hides provider selector component  or  ((Encounter.Provider <> FProvider) // and provider was changed (Note - no need to check User.DUZ here)  and (not DelayReviewChanges) // and no requests for delay of review  and ReviewChanges(TimedOut)) // Changes are valid  then  begin  if PersonFilter <> NPF\_SUPPRESS then // only change provider if it was requested  Encounter.Provider := FProvider;  Encounter.Location := FLocation;  Encounter.DateTime := FDateTime;  Encounter.VisitCategory := FVisitCategory;  Encounter.StandAlone := FStandAlone;  end;  end;  DelayReviewChanges := False;  DEAContext := False;  end;  finally  frmEncounter.Release;  end;  end; |
| Procedure  procedure TfrmEncounter.FormCreate(Sender: TObject); | Modified | Procedure was modified to setup updated control:  setUpSimilarNameSelector(cboPtProvider,aPr,''); |

Unit fNoteCPFields

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Constant | New | Defines message for cases of duplicate names  TX\_REQ\_AUTHOR\_DUPL = CRLF + 'The author confirmation failed.'; |
| Procedure  procedure EnterClinProcFields(ACPStatusFlag: integer; ErrMsg: string; var AProcSummCode: integer; var AProcDate: TFMDateTime; var AnAuthor: int64); | Modified | Procedure modified to update the initial setup of the cboAuthor.ItemName property with the value used as label on the screen:  cboAuthor.ItemName := copy(lblAuthor.Caption,1,Length(lblAuthor.Caption)-1); |
| Procedure  procedure TfrmNoteCPFields.FormCreate(Sender: TObject); | Modified | Procedure was modified to set initial values of properties of the updated TORComboBox components  procedure TfrmNoteCPFields.FormCreate(Sender: TObject);  begin  inherited;  FOKPressed := False;  cboAuthor.InitLongList(User.Name);  cboAuthor.SelectByIEN(User.DUZ);  setUpSimilarNameSelector(cboAuthor,aPr,'');  cboAuthor.msgNoSelection := TX\_REQ\_AUTHOR;  end; |
| Procedure  procedure TfrmNoteCPFields.ValidateFields(var ErrMsg: string); | Modified | Procedure was modified to use the properties of the updated component:  procedure TfrmNoteCPFields.ValidateFields(var ErrMsg: string);  begin  ErrMsg := cboAuthor.ValidationReport;  if (FCPStatusFlag = CP\_INSTR\_INCOMPLETE) then  begin  if cboProcSummCode.ItemIEN = 0 then ErrMsg := ErrMsg + TX\_REQ\_PROCSUMMCODE  else FProcSummCode := cboProcSummCode.ItemIEN;  if not calProcDateTime.IsValid then ErrMsg := ErrMsg + TX\_REQ\_PROCDATETIME  else if calProcDateTime.IsValid and (calProcDateTime.FMDateTime > FMNow) then ErrMsg := ErrMsg + TX\_NO\_PROC\_FUTURE  else FProcDateTime := calProcDateTime.FMDateTime;  end  else  begin  FProcSummCode := cboProcSummCode.ItemIEN;  if (calProcDateTime.FMDateTime > 0) then  begin  if (not calProcDateTime.IsValid) then ErrMsg := ErrMsg + TX\_INVALID\_PROCDATETIME  else if calProcDateTime.IsValid and (calProcDateTime.FMDateTime > FMNow) then ErrMsg := ErrMsg + TX\_NO\_PROC\_FUTURE  else FProcDateTime := calProcDateTime.FMDateTime;  end;  end;  ErrMsg := Trim(ErrMsg);  end; |

Unit fNoteProps

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Type TORComboBox | New | Type re-defined to use updated version of the TORComboBox  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Class Methods | New | Methods added to the type definition  procedure FormCreate(Sender: TObject);  procedure calNoteChange(Sender: TObject);  procedure cboNewTitleChange(Sender: TObject);  function getTitleIEN:Integer; |
| Function  function ExecuteNoteProperties(var ANote: TEditNoteRec; CallingTab: integer; IDNoteTitlesOnly,  IsNewIDEntry: boolean; AClassName: string; CPStatusFlag: integer): Boolean; | Modified | Function was modified to include setup for updated components:  // update messages for TORComboBoxex  cboAuthor.ItemName := copy(lblAuthor.Caption,1,Length(lblAuthor.Caption)-1);  cboCosigner.ItemName := copy(lblCosigner.Caption,1,Length(lblCosigner.Caption)-1);  Result := ShowModal = idOK; // display the form |
| Function  function TfrmNoteProperties.getTitleIEN:Integer; | New | Function returns value required for validation of input  function TfrmNoteProperties.getTitleIEN:Integer;  begin  if FDocType = TYP\_ADDENDUM then  Result := FAddend  else  begin  if cboNewTitle.ItemIEN = 0  then Result := FDocType  else Result := cboNewTitle.ItemIEN;  end;  end; |
| Procedure  procedure TfrmNoteProperties.cboNewTitleMouseClick(Sender: TObject); | Modified | Procedure was modified to track changes to FLastTitle value:  SetCosignerRequired(True);  FLastTitle := cboNewTitle.ItemIEN;  cboCosigner.TitleIEN := FLastTitle;  end; |
| Procedure  procedure TfrmNoteProperties.cboNewTitleExit(Sender: TObject); | Modified | Procedure was modified to track changes to FLastTitle value:  procedure TfrmNoteProperties.cboNewTitleExit(Sender: TObject);  begin  if cboNewTitle.ItemIEN <> FLastTitle then cboNewTitleMouseClick(Self);  cboCosigner.TitleIEN := FLastTitle;  end; |
| Procedure  procedure TfrmNoteProperties.cboNewTitleChange(Sender: TObject); | New | Procedure tracks changes to the TitleIEN  procedure TfrmNoteProperties.cboNewTitleChange(Sender: TObject);  begin  inherited;  cboCosigner.TitleIEN := cboNewTitle.ItemIEN;  end; |
| Procedure  procedure TfrmNoteProperties.cboAuthorExit(Sender: TObject); | Modified | Procedure was modified to use the updated version of control:  procedure TfrmNoteProperties.cboAuthorExit(Sender: TObject);  begin  inherited;  if cboAuthor.ItemIEN <> FLastAuthor then cboAuthorMouseClick(Self);  end; |
| Procedure  procedure TfrmNoteProperties.cmdOKClick(Sender: TObject); | Modified | Procedure was modified to use the updated version of TORComboBox component:  ErrMsg := ErrMsg + cboAuthor.ValidationReport; // similar provider validation  if not calNote.IsValid then ErrMsg := ErrMsg + TX\_REQ\_REFDATE;  if calNote.IsValid and (calNote.FMDateTime > FMNow) then ErrMsg := ErrMsg + TX\_NO\_FUTURE;  if cboCosigner.Visible then  begin  cboCosigner.TitleIEN := getTitleIEN;  ErrMsg := ErrMsg + cboCosigner.ValidationReport; // similar Cosigner validation  //if (cboCosigner.ItemIEN = User.DUZ) then ErrMsg := TX\_COS\_SELF; // (CanCosign will do this check) |
| Procedure  procedure TfrmNoteProperties.FormCreate(Sender: TObject); | New | Procedure sets properties of the updated components:  procedure TfrmNoteProperties.FormCreate(Sender: TObject);  begin  inherited;  setUpSimilarNameSelector(cboAuthor,aPr,’’);  cboAuthor.msgNoSelection := TX\_REQ\_AUTHOR;  setUpSimilarNameSelector(cboCosigner,aCo,’’);  cboCosigner.msgNoSelection := TX\_REQ\_COSIGNER  end; |
| Procedure  procedure TfrmNoteProperties.calNoteChange(Sender: TObject); | New | Procedure tracks changes to the Validate Date:  procedure TfrmNoteProperties.calNoteChange(Sender: TObject);  begin  inherited;  cboCosigner.ValidationDate := calNote.FMDateTime;  end; |

Unit fPtSel

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Function  function TfrmPtSel.DupLastSSN(var DFN: Int64): Boolean; | Modified | Function was modified to use updated version of patient selector:  function TfrmPtSel.DupLastSSN(var DFN: Int64): Boolean;  var  i: Integer;  SL: TStrings;  const  fmtResultError = 'SubsetOfPatientsWithSimilarSSNs returns incorrect data.' +  CRLF + CRLF + 'Search for DFN %d returns' + CRLF + ' %s' ;  begin  Result := False;  SL := SubsetOfPatientsWithSimilarSSNs(DFN);  if SL.Count > 0 then  begin  if SL[0] = '1' then  begin  Result := True;  SL.Delete(0);  i := 0;  while i < SL.Count do  if Piece(SL[i], U, 1) = '1' then  begin  SL[i] := (Piece(SL[i], U, 2) + U + Piece(SL[i], U, 3) + U +  FormatFMDateTimeStr('mmm dd,yyyy', Piece(SL[i], U, 4)) + U +  Piece(SL[i], U, 5));  inc(i)  end  else  SL.Delete(i);  case SL.Count of  0:;  1: begin  if Piece(SL[0],U,1) <> IntToStr(DFN) then  MessageDlg(Format(fmtResultError,[DFN,SL[0]]),mtError,[mbOK],0)  else  ;  end;  else // Call form to get user's selection from expanded duplicate pt. list  DFN := getItemIDFromList(SL);  end;  end;  end;  SL.Free;  end; |
|  |  |  |

Unit fODConsult

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Type TORComboBox | New | Type re-defined to use updated version of the TORComboBox  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Form component renamed to  cboAttn | Modified | Original name txtAttn did not match the component type of TORComboBox. The component was renamet to cboAttn |
| Procedure  procedure TfrmODCslt.FormCreate(Sender: TObject); | Modified | Procedure was modified to include setup for updated component  consultQuickOrder := false;  setUpSimilarNameSelector(cboAttn,aPr,'');  end; |
| Procedure  procedure TfrmODCslt.Validate(var AnErrMsg: string); | Modified | Procedure was modified to use methods of the updated component for validation:  if (lblClinicallyIndicated.Enabled) and (calClinicallyIndicated.FMDateTime < FMToday) then SetError(TX\_PAST\_DATE);  anErrMsg := anErrMsg + cboAttn.ValidationReport;  AnErrMsg := Trim(AnErrMsg);  end; |
| Procedure  procedure TfrmODCslt.ControlChange(Sender: TObject); | Modified | Procedure was modified to track changes to cboAttn component:  procedure TfrmODCslt.ControlChange(Sender: TObject);  var  x: string;  i: integer;  begin  inherited;  cboAttn.NeedsValidation := Sender = cboAttn;  … |
| Procedure  procedure TfrmODCslt.cmdAcceptClick(Sender: TObject); | Modified | Procedure modified to ensure execution of all methods required for validation:  procedure TfrmODCslt.cmdAcceptClick(Sender: TObject);  var  BADiagnosis: string;  begin  inherited;  cmdAccept.SetFocus; // NSR#20110606 - making sure we exit other controls  if treService.Selected <> nil then  LastNode := treService.Selected.AbsoluteIndex;  … |

Unit fODProc

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Type TORComboBox | New | Type re-defined to use updated version of the TORComboBox  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Form component renamed to  cboAttn | Modified | Original name txtAttn did not match the component type of TORComboBox. The component was renamet to cboAttn |
| Procedure  procedure TfrmODCslt.FormCreate(Sender: TObject); | Modified | Procedure was modified to include setup for updated component  consultQuickOrder := false;  setUpSimilarNameSelector(cboAttn,aPr,'');  end; |
| Procedure  procedure TfrmODCslt.Validate(var AnErrMsg: string); | Modified | Procedure was modified to use methods of the updated component for validation:  if (lblClinicallyIndicated.Enabled) and (calClinicallyIndicated.FMDateTime < FMToday) then SetError(TX\_PAST\_DATE);  anErrMsg := anErrMsg + cboAttn.ValidationReport;  AnErrMsg := Trim(AnErrMsg);  end; |
| Procedure  procedure TfrmODCslt.ControlChange(Sender: TObject); | Modified | Procedure was modified to track changes to cboAttn component:  procedure TfrmODCslt.ControlChange(Sender: TObject);  var  x: string;  i: integer;  begin  inherited;  cboAttn.NeedsValidation := Sender = cboAttn;  … |

Unit fOrdersAlert

Unit was modified to avoid using of custom flags for closing of modal dialog. The original TORComboBox was replaced with the updated version and used for validation of the selected name

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Type TORComboBox | New | Type re-defined to use updated version of the TORComboBox  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Form methods | Removed | The next methods were removed:  procedure cmdOKClick(Sender: TObject);  procedure cmdCancelClick(Sender: TObject);  procedure cboOnExit(Sender: TObject); |
| Form field | Removed | Field removed as not used for validation of the result  OKPressed:Boolean |
| Procedure  function ExecuteAlertOrders(SelectedList: TList): Boolean; | Modified | Procedure was modified to use updated control  function ExecuteAlertOrders(SelectedList: TList): Boolean;  var  frmAlertOrders: TfrmAlertOrders;  i: Integer;  AlertRecip: Int64;  begin  Result := False;  if SelectedList.Count = 0 then  Exit;  frmAlertOrders := TfrmAlertOrders.Create(Application);  try  ResizeFormToFont(TForm(frmAlertOrders));  frmAlertOrders.SetUp(SelectedList);  if frmAlertOrders.ShowModal = mrOK then  begin  AlertRecip := frmAlertOrders.cboAlertRecipient.ItemIEN;  for i := 0 to SelectedList.Count - 1 do  AlertOrder(TOrder(SelectedList.Items[i]), AlertRecip);  Result := True;  end;  finally  frmAlertOrders.Release;  for i := 0 to SelectedList.Count - 1 do  UnlockOrder(TOrder(SelectedList.Items[i]).ID);  end;  end; |
| Procedure  procedure TfrmAlertOrders.FormCloseQuery(Sender: TObject; | New | Procedure is used for validation of the user input;  procedure TfrmAlertOrders.FormCloseQuery(Sender: TObject;  var CanClose: Boolean);  var  ErrMsg: String;  begin  inherited;  if ModalResult = mrOK then  begin  ErrMsg := cboAlertRecipient.ValidationReport;  CanClose := ErrMsg = '';  if not CanClose then  InfoBox(ErrMsg, 'Validation Error', MB\_OK);  end;  end; |
| Procedure  procedure TfrmAlertOrders.FormCreate(Sender: TObject); | Modified | Procedure was modified to use the updated control  procedure TfrmAlertOrders.FormCreate(Sender: TObject);  begin  inherited;  setUpSimilarNameSelector(cboAlertRecipient, aPr, '');  cboAlertRecipient.ItemName := copy(lblAlertRecipient.Caption, 1,  Length(lblAlertRecipient.Caption));  end; |
| Procedure  procedure TfrmAlertOrders.FormKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState); | New | Procedure lets close the dialog by Esc button  procedure TfrmAlertOrders.FormKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if Key = VK\_ESCAPE then  ModalResult := mrCancel;  end; |
| Procedure  procedure TfrmAlertOrders.SetUp(aList: TList); | New | Procedure sets up the updated components  procedure TfrmAlertOrders.SetUp(aList: TList);  var  i: Integer;  AnOrder: TOrder;  AlertRecip: Int64;  Provider: String;  begin  if not assigned(aList) then  Exit;  AnOrder := TOrder(aList.Items[0]); // use first order's provider  Provider := AnOrder.ProviderName;  AlertRecip := AnOrder.Provider;  cboAlertRecipient.InitLongList(Provider);  cboAlertRecipient.SelectByIEN(AlertRecip);  for i := 0 to aList.Count - 1 do  lstOrders.Items.Add(TOrder(aList.Items[i]).Text);  end; |

Unit fPtSelOptns

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Procedure  procedure TfrmPtSelOptns.cboListDblClick(Sender: TObject); | New | Procedure created to validate names on double click of the component  procedure TfrmPtSelOptns.cboListDblClick(Sender: TObject);  var  DFN : Int64;  begin  inherited;  DFN := getPersonCheckedForSimilarNames(cboList.ItemIEN,'PROVIDER');  if (DFN > -1) then  begin  cboList.SelectByIEN(DFN);  FSetPtListTop(DFN);  end;  end; |
| Procedure  procedure TfrmPtSelOptns.SetDefaultPtList(Dflt: string); | Modified | Procedure modified to adjust position of the form component  procedure TfrmPtSelOptns.SetDefaultPtList(Dflt: string);  procedure AdjustRadioButtons(aVisible:Boolean);  var  cmp: TControl;  i, off: Integer;  begin  if aVisible then  exit;  // move buttons up if radDflt is hidden  off := radProviders.Top - radDflt.Top;  // bvlPtList.Height := bvlPtList.Height - off;  for i := 0 to bvlPtList.ControlCount -1 do  begin  cmp := bvlPtList.Controls[i];  if (cmp is TRadioButton) and (cmp <> radDflt) then  TRadioButton(cmp).Top := TRadioButton(cmp).Top - off;  end;  bvlPtList.Invalidate;  Application.ProcessMessages;  end;  begin  if Length(Dflt) > 0 then // if default patient list available, use it  begin  radDflt.Caption := '&Default: ' + Dflt;  radDflt.Checked := True; // causes radHideSrcClick to be called  end  else // otherwise, select from all patients  begin  radDflt.Enabled := False;  radDflt.Visible := not ScreenReaderActive;  radAll.Checked := True; // causes radHideSrcClick to be called  bvlPtList.TabStop := True;  bvlPtList.Hint := 'No default radio button unavailable 1 of 7 to move to the other patient list categories press tab';  // fixes CQ #4716: 508 - No Default rad btn on Patient Selection screen doesn't read in JAWS. [CPRS v28.1] (TC).  adjustRadioButtons(radDflt.Enabled);  end;  end; |

Unit rCore

Functions added supporting search for providers/cosigners/persons with similar names

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Function  function SubsetOfActiveAndInactivePersonsWithSimilarNames(aDUZ:Int64;aKey: string): TStrings; | New | function SubsetOfActiveAndInactivePersonsWithSimilarNames(aDUZ:Int64;aKey: string): TStrings;  { returns a pointer to a list of users holding key aKey and names similar to the one  with provided DUZ }  begin  Result := TStringList.create;  tCallV(Result,'ORWU NEWPERS',[aDUZ,1,aKey,'','','',True]);// TRUE - to indicate this is NSR 20110606  end; |
| Function  SubsetOfCosignersWithSimilarNames(aDUZ:Int64;CSPDate:Double;TITLEIEN:Integer): TStrings; | New | function SubsetOfCosignersWithSimilarNames(aDUZ:Int64;CSPDate:Double;TITLEIEN:Integer): TStrings;  { returns a pointer to a list of users that can sign document with TITLEIEN and names similar to the one  with provided DUZ }  begin  Result := TStringList.create;  tCallV(Result,'ORWU2 COSIGNER',[aDUZ,1,CSPDate,0,TITLEIEN,True]);  end; |
| Function  function SubsetOfPatientsWithSimilarSSNs(aDFN:Int64): TStrings; | New | function SubsetOfPatientsWithSimilarSSNs(aDFN:Int64): TStrings;  { returns a pointer to a list of patients that has similar SSNs}  begin  Result := TStringList.create;  tCallV(Result,'DG CHK BS5 XREF ARRAY', [aDFN]);  end; |

Unit uEditConsult

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Type TORComboBox | New | Type re-defined to use updated version of the TORComboBox  type  // names validation support  TORComboBox = class(uSimilarNames.TORComboBox)  end; |
| Form component renamed to  cboAttn | Modified | Original name txtAttn did not match the component type of TORComboBox. The component was renamet to cboAttn |
| Procedure  procedure TfrmODCslt.FormCreate(Sender: TObject); | New | Procedure created to include setup for updated component  procedure TfrmEditCslt.FormCreate(Sender: TObject);  begin  inherited;  setUpSimilarNameSelector(cboAttn,aPr,'');  end; |
| Procedure  procedure TfrmODCslt.Validate(var AnErrMsg: string); | Modified | Procedure was modified to use methods of the updated component for validation:  anErrMsg := anErrMsg + cboAttn.ValidationReport;  AnErrMsg := Trim(AnErrMsg);  end; |
| Procedure  procedure TfrmODCslt.ControlChange(Sender: TObject); | Modified | Procedure was modified to track changes to cboAttn component:  procedure TfrmEditCslt.ControlChange(Sender: TObject);  begin  if FChanging then exit;  if Sender = cboAttn then  cboAttn.NeedsValidation := True;  … |
|  |  |  |
|  |  |  |

Unit fVisitType

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |
| Procedure  procedure TfrmVisitType.btnAddClick(Sender: TObject); | Modified | Procedure was modified to add validation of the providers with similar names  procedure TfrmVisitType.btnAddClick(Sender: TObject);  var  DUZ: Int64;  begin  inherited;  DUZ := getProviderIdCheckedForSimilarName(cboPTProvider.ItemIEN,'PROVIDER',lbProviders.Items);  if DUZ > 0 then  begin  cboPTProvider.SelectByIEN(DUZ);  uProviders.AddProvider(IntToStr(cboPTProvider.ItemIEN), cboPTProvider.Text, FALSE);  RefreshProviders;  lbProviders.SelectByIEN(cboPTProvider.ItemIEN);  end;  end; |
| Procedure  procedure TfrmVisitType.cboPtProviderKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState); | New | Procedure added to allow adding the names by clicking the Return;  procedure TfrmVisitType.cboPtProviderKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if Key = VK\_RETURN then  btnAddClick(nil);  end; |

Unit uSimilarNames

This unit defines the changes to TORComboBox component.

The enhanced component is used for validation of the selected item.

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Type  TSelector | New | The type defines several versions of selector  TSelector = (APt, APr, ACo, Unknown); |
|  |  |  |
| Type TORComboBox | New | New type created to support validation of the selected item |
| Private field  fSelectorType:TSelector | New | Field identifies type of the selector |
| Private field  fFilter:String | New | Value to use as filter when requesting data from the server |
| Private field  fNeedValidation:Boolean | New | Flag is reset to “True” on every change of the selected item. Set to “False” after the user confirms selection. |
| Private procedure  procedure OnComponentChange(Sender: TObject); | New | Procedure resets fNeedsValidation flag every time the new item is selected  procedure TORComboBox.OnComponentChange(Sender: TObject);  begin  inherited;  fNeedsValidation := True;  end; |
| Private procedure  procedure OnComponentKeyDown(Sender: TObject; var Key: Word; | New | Procedure implements validation of the selected name when user clicks Return key  procedure OnComponentKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if Key = VK\_Return then  begin  if NeedsValidation then  begin  if isValidSelection(Sender) then  fNeedsValidation := False;  Key := 0; // to prevent the default button to close the paren dialog  end;  end;  end; |
| Private procedure  procedure OnComponentDblClick(Sender: TObject); | New | Procedure implements validation of the selected name when user double clicks the selected item  procedure TORComboBox.OnComponentDblClick(Sender: TObject);  begin  inherited;  if NeedsValidation then  if isValidSelection(Sender) then  fNeedsValidation := False;  end; |
| Private procedure  procedure setSelectorType(aType: TSelector); | New | Property setter:  procedure TORComboBox.setSelectorType(aType: TSelector);  begin  fSelectorType := aType;  end; |
| Public field  ItemName:String | New | Field is used to generate information messages |
| Public field  Exceptions:TStrings | New | The list of exceptions – names included in the list should not be allowed for selection. |
| Public field  ValidationDate:Double | New | Value of Date to use for validation of the document cosigner |
| Public field  TitleIEN:String | New | Value to use as the Note IEN for validation of the note cosigner |
| Public field  msgNoSelection:String | New | Value to use as the message in case the selection was not done |
| Public Property  NeedsValidation:Boolean | New | Flag identifying need in validation of the selected item |
| Public property  SelectorType | New | Property identifying the version of the selector |
| Constructor  Create | New | Constructor includes assignment for the updated components:  constructor TORComboBox.Create(AOwner: TComponent);  begin  inherited Create(AOwner);  OnChange := OnComponentChange;  OnKeyDown := OnComponentKeyDown;  OnDblClick := OnComponentDblClick;  NeedsValidation := True;  SelectorType := Unknown;  end; |
| Function  function isValidSelection(Sender: TObject): Boolean; | New | Function implements validation of the form prior to closing  function TORComboBox.isValidSelection(Sender: TObject): Boolean;  begin  Result := False;  if ItemIEN < 0 then  exit;  case SelectorType of  APt:  ;  APr:  Result := getProviderIdCheckedForSimilarName(ItemIEN, fFilter) > 0;  ACo:  Result := getCosignerIdCheckedForSimilarNames(ItemIEN, ValidationDate,  TitleIEN) > 0;  Unknown:  ;  end;  end; |
| Function  function ValidationReport: String; | New | Validation report is used by calling components to check if the component validated  function TORComboBox.ValidationReport: String;  var  errMsg: String;  begin  case fSelectorType of  Unknown:  Result := 'Unknown type of validation';  APt:  Result := 'Patient Validation';  APr:  begin  if ItemIEN = 0 then  Result := msgNoSelection  else if (not NeedsValidation) or isSingleProviderDUZ(ItemIEN, fFilter,  errMsg, ItemName) then  Result := ''  else  Result := errMsg;  end;  ACo:  begin  if ItemIEN = 0 then  Result := msgNoSelection  else if (not NeedsValidation) or isSingleCosignerDUZ(ItemIEN,  ValidationDate, TitleIEN, errMsg, ItemName) then  Result := ''  else  Result := errMsg;  end;  end;  if Result <> '' then  try  Result := CRLF + Result;  SetFocus;  except  on E: Exception do  ShowMessage('ERROR (TORComboBox.ValidationReport):' + CRLF + E.Message);  end;  end; |
| Procedure  function getItemIDFromList(aList: TStrings; aType: TSelector = APt;  anExceptions: TStrings = nil): Int64; | New | General selection function  function getItemIDFromList(aList: TStrings; aType: TSelector = APt;  anExceptions: TStrings = nil): Int64;  var  frmDupPts: TfrmDupPts;  begin  Result := -1;  if assigned(aList) then  begin  frmDupPts := TfrmDupPts.CreateSelector(aType, aList, anExceptions);  try  if frmDupPts.ShowModal = mrOK then  Result := frmDupPts.lboSelPt.ItemID;  finally  frmDupPts.Release;  end; |
| Function  function isSingleProviderDUZ(aDUZ: Int64; aKey: String; var msg: String;  ItemName: String = 'Provider'): Boolean; | New | Function validates if the provided DUZ matches single provoder  function isSingleProviderDUZ(aDUZ: Int64; aKey: String; var msg: String;  ItemName: String = 'Provider'): Boolean;  var  SL: TStrings;  begin  Result := False;  SL := SubsetOfActiveAndInactivePersonsWithSimilarNames(aDUZ, aKey);  try  case SL.Count of  0:  msg := Format(fmtInvalidItemSelected, [ItemName]);  1:  Result := True;  else  msg := Format(fmtMultipleItemNames, [ItemName]);  end;  finally  SL.Free;  end;  end; |
| Function  function isSingleCosignerDUZ(aDUZ: Int64; CSPDate: Double; TitleIEN: Integer;  var msg: String; ItemName: String = 'Cosigner'): Boolean; | New | Function validates if the provided DUZ matches single provoder  function isSingleCosignerDUZ(aDUZ: Int64; CSPDate: Double; TitleIEN: Integer;  var msg: String; ItemName: String = 'Cosigner'): Boolean;  var  SL: TStrings;  begin  Result := False;  SL := nil;  try  SL := SubsetOfCosignersWithSimilarNames(aDUZ, CSPDate, TitleIEN);  if not assigned(SL) or (SL.Count = 0) then  msg := Format(fmtInvalidItemSelected, [ItemName])  else if SL.Count > 1 then  msg := Format(fmtMultipleItemNames, [ItemName])  else  Result := True;  finally  if assigned(SL) then  SL.Free;  end;  end; |
| Function  function getPersonCheckedForSimilarNames(aDUZ: Int64; aKey: String = '';  anExceptions: TStrings = nil): Int64; | New | Function provides DUZ of the person checked for similar names  function getPersonCheckedForSimilarNames(aDUZ: Int64; aKey: String = ''; anExceptions: TStrings = nil): Int64;  var  SL: TStrings;  sDUZ: String;  function countExceptions: Integer;  var  sDUZ: String;  i, j: Integer;  begin  Result := 0;  if not assigned(anExceptions) then exit;  i := 0;  while i < SL.Count do  begin  sDUZ := Piece(SL[i], U, 1);  for j := 0 to anExceptions.Count - 1 do  if Piece(anExceptions[j], U, 1) = sDUZ then  begin  Inc(Result);  break;  end;  Inc(i);  end;  end;  begin  Result := -1;  SL := SubsetOfActiveAndInactivePersonsWithSimilarNames(aDUZ, aKey);  try  case SL.Count of  0: ShowMessage(Format(fmtInvalidItemSelected, ['']) +  ' Please Select another name');  1: begin  sDUZ := Piece(SL[0], U, 1);  if sDUZ = IntToStr(aDUZ) then  Result := aDUZ  else  MessageDLG('getPersonCheckedForSimilarNames:' + #13#10#13#10 +  'Search for DUZ=' + IntToStr(aDUZ) +  ' returns one record with DUZ=' + sDUZ, mtError, [mbOK], 0);  end  else begin  if SL.Count = countExceptions then  MessageDLG  ('The selected person and all the persons with similar names were already added'  + CRLF + 'Please select another person', mtInformation, [mbOK], 0)  else  Result := getItemIDFromList(SL, APr, anExceptions);  end;  end;  finally  SL.Free;  end;  end; |
| Function  function getProviderIdCheckedForSimilarName(aDUZ: Int64; aKey: String = '';  anExceptions: TStrings = nil): Int64; | New | Function provides DUZ of the provider checked for similar names  function getProviderIdCheckedForSimilarName(aDUZ: Int64; aKey: String = '';  anExceptions: TStrings = nil): Int64;  begin  // "Similar Provider" window should not be shown if the logged on user is the selected provider  if aDUZ = User.DUZ then  Result := aDUZ  else  Result := getPersonCheckedForSimilarNames(aDUZ, aKey, anExceptions);  end; |
| Function  function getCosignerCheckedForSimilarNames(aDUZ: Int64; CSPDate: Double;  TitleIEN: Integer): Int64; | New | Function provides DUZ of the cosigner checked for similar names  function getCosignerCheckedForSimilarNames(aDUZ: Int64; CSPDate: Double;  TitleIEN: Integer): Int64;  var  sDUZ: String;  SL: TStrings;  begin  Result := -1;  SL := SubsetOfCosignersWithSimilarNames(aDUZ, CSPDate, TitleIEN);  case SL.Count of  {$IFDEF DEBUG\_AA}  0:  ShowMessage(Format('getCosignerCheckedForSimilarNames' + #13#10 +  fmtInvalidDUZ, [aDUZ]));  {$ELSE}  0:  ShowMessage(Format(fmtInvalidItemSelected, ['']) +  ' Please Select another name');  {$ENDIF}  1:  begin  sDUZ := Piece(SL[0], U, 1);  if sDUZ = IntToStr(aDUZ) then  Result := aDUZ  else  MessageDLG('getCosignerCheckedForSimilarNames:' + #13#10#13#10 +  'Search for DUZ=' + IntToStr(aDUZ) + ' returns one record with DUZ='  + sDUZ, mtError, [mbOK], 0);  end  else  Result := getItemIDFromList(SL, ACo);  end;  SL.Free;  end; |
| Function  function getCosignerIdCheckedForSimilarNames(aDUZ: Int64; CSPDate: Double;  TitleIEN: Integer): Int64; | New | Function Cosigner DUZ of the checked for similar names  function getCosignerIdCheckedForSimilarNames(aDUZ: Int64; CSPDate: Double;  TitleIEN: Integer): Int64;  begin  // "Similar Provider" window should not be shown if the logged on user is the selected provider  if aDUZ = User.DUZ then  Result := aDUZ  else  Result := getCosignerCheckedForSimilarNames(aDUZ, CSPDate, TitleIEN);  end; |
| Procedure  procedure setUpSimilarNameSelector(aSelector: TORComboBox; aType: TSelector;  aFilter: String); | New | Procedure sets up different types of the selector  procedure setUpSimilarNameSelector(aSelector: TORComboBox; aType: TSelector;  aFilter: String);  begin  if aSelector = nil then  exit;  with aSelector do  begin  SelectorType := aType;  fFilter := aFilter;  end;  end; |
| Constants | New | Constants are used to format messages produced by the component  const  fmtInvalidItemSelected =  'The name selected is not a CPRS user name allowable for entry in this %s field.';  fmtMultipleItemNames =  'The %s name selected is not unique. The name confirmation is required.'; |

Unit CPRSChart.dpr

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Interface uses clause | Modified | Reference on uSimilarNames added |

###### Update Surrogate Management Functionality within CPRS GUI (20071216)

Summary of changes

1. The new tab added to the Options dialog. The new Tab “Surrogates” provides information on the currently assigned surrogates and allows user modify the surrogates settings (file fOptions)
2. The form created for presentation of the surrogates’ settings (file fOptionsSurrogates.pas). The form encapsulates functionality related to the surrogate management.
3. New dialog created to allow user select surrogate and date range the surrogate is assigned (file fSurrogateEdit.pas)
4. TORCalendar class has been modified to implement range checking (file ORDtTm.pas )
5. TORDateTimeDlg class has been modified to implement range checking (file ORDtTm.pas )
6. TORDateBox class has been modified to implement range checking (file ORDtTm.pas )

Files modified

|  |  |
| --- | --- |
| CPRSChart.dpr | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  Project file updated with references to the new units |
| fOptions.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  Added “Surrogates” tab to let user manage surrogates settings. The tab uses TPanel object (pnlSurrogates) as the container for TfrmOptionsSurrogate object that implements surrogates management functionality.  The form TfrmOptions.FormCreate procedure updated to create the TfrmOptionsSurrogates object and assign pnlSurrogates as its parent. |
| fOptionsSurrogate.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  New form added to the project. The form implements surrogates management functionality. |
| fSurrogateEdit.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  The dialog form was updated. Class TfrmSurrogateEdit created to implement selection of the surrogate settings – surrogate name and date range. |
| uConst.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  Constant definition UM\_CHECKAPPLY (WM\_USER + 9600) add. This constant identified message sent to notify application on changes to the surrogates settings. |
| rOptions.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  New procedure rpcGetSurrogateInfoList added. The procedure provides user’s surrogates settings. |
| ORDtTm.pas | **RSD #2.6.3 Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request (#20071216)**  Unit updated to implement range check in TORCalendar object |

GUI Classes

File CPRSChart.dpr changes

| Program | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| CPRSChart | File CPRSChart.dpr  Modifications to implement Multiple surrogates management functionality (Saving/restoring surrogates settings to/from server) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Units Added** | **fOptionsSurrogate in 'Options\fOptionsSurrogate.pas' {frmOptionsSurrogate},**  **fSurrogateEdit in 'Options\fSurrogateEdit.pas' {frmSurrogateEdit},**  **ORDtTm in 'ORDtTm.pas' {ORfrmDtTm},** | | | |

File fOptions.pas changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| TfrmOptions | fOptions.pas  Added “Surrogates” tab to let user manage surrogates settings. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| Controls added | **pnlSurogates: TPanel;**  **pnlSurrogatesTop: TPanel**  **Bevel2: TBevel;**  **StaticText2: TStaticText;**  **StaticText3: TStaticText;** | | | |
| Method(s) added | **procedure MSG\_CHECKAPPLY(var Message: TMessage); message UM\_CHECKAPPLY;**  **procedure setFormParented(aForm:TForm; aParent:TWinControl;anAlign: TAlign = alClient);** | | | |
| Method(s) modified | **procedure TfrmOptions.FormCreate(Sender: TObject);** | | | |
| Unit(s) references added | **uConst , fOptionsSurrogate** | | | |
| **Class TfrmOptions Modified Logic (New code added)** | | | | |
| **procedure TfrmOptions.MSG\_CHECKAPPLY(var Message: TMessage);**  **begin**  **CheckApply;**  **end;**  **procedure setFormParented(aForm:TForm; aParent:TWinControl;anAlign: TAlign = alClient);**  **begin**  **if aForm.Parent <> aParent then**  **begin**  **aForm.BorderStyle := bsNone;**  **aForm.Parent := aParent;**  **aForm.Align := anAlign;**  **aForm.Menu := nil;**  **aForm.Show;**  **end;**  **end;** | | | | |
| **Class TfrmOptions Modified Logic (New code added is shown in bold)** | | | | |
| procedure TfrmOptions.FormCreate(Sender: TObject);  // initialize form  **// NSR20071216 AA 2016-01-23 ------------------------------------------- begin**  **procedure setSurrogates;**  **begin**  **Application.CreateForm(TfrmOptionsSurrogate,frmOptionsSurrogate);**  **setFormParented(frmOptionsSurrogate,pnlSurrogates);**  **end;**  **// NSR20071216 AA 2016-01-23 --------------------------------------------- end**  begin  LoadNotifications;  LoadOrderChecks;  FdirtyNotifications := false;  FdirtyOrderChecks := false;  FdirtyOtherStuff := false;  **setSurrogates; // NSR20071216 AA 2016-01-23**  CheckApply;  if (Encounter.Provider = 0) and not IsCIDCProvider(User.DUZ) then  btnDiagnoses.Enabled := False;  FGiveMultiTabMessage := ScreenReaderSystemActive;  end; | | | | |

|  |
| --- |
| **Class TfrmOptions** Current Form |
| Original Layout of Notification tab of the Options dialog |
| **Class TfrmOptions** Modified Form |
| Updated layout of the "Notifications" tab of the Options dialog.  New Tab:  Updated layout of the "Notifications" tab of the Options dialogincludes "prcessed Alerts Settings..." button |

File fOptionsSurrogate.pas

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **TfrmOptionsSurrogate** | File fOptionsSurrogate.pas  Implements TfrmOptionsSurrogate form | | | |
| Description | Unit was completely redesigned to implement surrogate data table and assignment management.  The updated form encapsulates management functionality: presentation of the surrogate assigned to the user, buttons to initiate assignment, editing or deletion of the assignment records.  The form supposed to be used as the “child” component of the Options dialog (pnlSurrogates used as the parent of the TfrmOptionsSurrogate) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| Form layout | Surrogate table includes information on surrogate assignments for the user. | | | |
| **Class TfrmOptionsSurrogate Modified Logic (New code added)** | | | | |
| See files fOptionsSurrogate.pas; fOptionsSurrogate.dfm | | | | |

Components on Form TfrmOptionsSurrogate

| Name | Type | Description |
| --- | --- | --- |
| clvSurrogates | TCaptionListView | List view presenting surrogates assignments. |
| pnlDebug | TPanel | Debug information presenter |
| pnlToolBar | TPanel | Container of form action buttons |
| pnlUpdateIndicator | Tpanel | Container of change indicator object |
| stxtChanged | TstaticText | Indicator of changes |
| pnlInfo | Tpanel | Informational message container |
| pnlSurrogateTools | Tpanel | Top level container of the form components |
| VA508StaticText1 | TVA508StaticText | Informational message presenter |
| btnRemove | TButton | Removes the selected assignment. |
| btnSurrEdit | TButton | Edits the surrogate setting or adds a new one |
| ckbDebug | TCheckbox | Controls format of debug information |

Event Handlers of form TfrmOptionsSurrogate

| Method Name | Procedure / Function | Description |
| --- | --- | --- |
| procedure btnRemoveClick | procedure | Implements removal of the assigned surrogate  procedure TfrmOptionsSurrogate.btnRemoveClick(Sender: TObject);  var  i: Integer;  sMsg: String;  begin  i := clvSurrogates.ItemIndex;  if i < 0 then  exit;  if not assigned(clvSurrogates.Items[clvSurrogates.ItemIndex].Data) then  ShowMessage('Item should have object assigned!')  else  begin  sMsg := TIDItem(clvSurrogates.Items[clvSurrogates.ItemIndex].Data).getInfo;  if MessageDlg(msgSurrogateRemove + CRLF + CRLF + sMsg, mtConfirmation,  [mbOK, mbCancel], 0) <> mrOK then  exit;  clvSurrogates.Items.BeginUpdate;  clvSurrogates.Items.Delete(i);  clearRanges;  setRangeInfo;  reNumItems;  clvSurrogates.Items.EndUpdate;  SurrogateUpdated := fRawServerData <> ListViewToRaw; // True;  setByCurrentItem;  end;  end; |
| procedure clvSurrogatesCompare | procedure | Used for sorting of the assignments by “From” date  procedure TfrmOptionsSurrogate.clvSurrogatesCompare(Sender: TObject;  Item1, Item2: TListItem; Data: Integer; var Compare: Integer);  var  dt1, dt2: TDateTime;  begin  inherited;  dt1 := 0.0;  dt2 := 0.0;  if assigned(Item1.Data) then  dt1 := TIDItem(Item1.Data).dateFrom;  if assigned(Item2.Data) then  dt2 := TIDItem(Item2.Data).dateFrom;  Compare := 0;  if dt1 < dt2 then  Compare := -1  else if dt2 < dt1 then  Compare := 1;  end; |
| procedure clvSurrogatesCustomDrawItem | procedure | Customizes presentation of assigned records  procedure TfrmOptionsSurrogate.clvSurrogatesCustomDrawItem  (Sender: TCustomListView; Item: TListItem; State: TCustomDrawState;  var DefaultDraw: Boolean);  begin  inherited;  if Item.Caption = ciOpen then  Sender.Canvas.Font.Color := cl3dDkShadow // clHighlight  else  Sender.Canvas.Font.Color := clWindowText; // clDkGray  end; |
| procedure clvSurrogatesChange | procedure | Controls availability of buttons according to the selected record  procedure TfrmOptionsSurrogate.clvSurrogatesChange(Sender: TObject;  Item: TListItem; Change: TItemChange);  begin  if fIgnore then  exit;  inherited;  if (Sender as TListView).ItemIndex = -1 then  btnSurrEdit.Enabled := (Sender as TListView).Items.Count > 0  else  begin  fIgnore := True;  btnSurrEdit.Enabled := True;  setByCurrentItem;  fIgnore := False;  end;  {$IFDEF DEBUG\_AA}  pnlDebug.Caption := TIDItem(Item.Data).getInfoDebug;  {$ENDIF}  end; |
| procedure btnSurrEditClick | procedure | Implements Add or Edit action  procedure TfrmOptionsSurrogate.btnSurrEditClick(Sender: TObject);  var  IDNext, IDPrev, IDItem: TIDItem;  i: LongInt;  bLastRecord: Boolean;  sDataOld, sData, sComment, sSurrogate, sStart, sStop, sMin, sMax: String;  begin  inherited;  if clvSurrogates.Items.Count < 1 then  {$IFDEF DEBUG}  ShowMessage('Surrogates list should not be blank... :( ')  {$ENDIF}  else  begin  i := clvSurrogates.ItemIndex;  bLastRecord := i = clvSurrogates.Items.Count - 1;  if i < 0 then  i := 0;  IDItem := TIDItem(clvSurrogates.Items[i].Data);  if not assigned(IDItem) then  begin  ShowMessage('No item to process...');  exit;  end;  if i > 0 then  IDPrev := TIDItem(clvSurrogates.Items[i - 1].Data)  else  IDPrev := nil;  if i < clvSurrogates.Items.Count - 1 then  IDNext := TIDItem(clvSurrogates.Items[i + 1].Data)  else  IDNext := nil;  sStart := IDItem.strFrom;  sStop := IDItem.strUntil;  sSurrogate := IDItem.PublicName;  if (i > 0) and assigned(IDPrev) and IDPrev.IsOpen then  sMin := IDPrev.strFrom  else  sMin := IDItem.strFrom;  if StrDateToDate(sMin) < Now() then  sMin := FormatDateTime(fmtDateTime, Now);  if (i < clvSurrogates.Items.Count - 1) and assigned(IDNext) and IDNext.IsOpen  then  sMax := IDNext.strUntil  else  sMax := IDItem.strUntil;  sData := getItemID(i);  sDataOld := sData;  if editSurrogate(sSurrogate, sStart, sStop, sMin, sMax, sData, bLastRecord)  = mrOK then  begin  {$IFDEF DEBUG\_AA}  ShowMessage('Edit Surrogate Results: ' + CRLF + CRLF + 'Name: ' +  sSurrogate + '(' + sData + ')' + CRLF + 'Start: ' + sStart + CRLF +  'Stop: ' + sStop + CRLF);  {$ENDIF}  sComment := '';  if i = 0 then  sComment := ciActive;  if sDataOld <> sData then  begin  IDItem := newTIDItem(fIDCollection, sData, '', sSurrogate, sComment,  StrDateToDate(sStart), StrDateToDate(sStop));  end  else  begin  IDItem.IDString := sData;  IDItem.PublicName := sSurrogate;  IDItem.dateFrom := FMDateTimeToDateTime(StrToFloat(sStart));  IDItem.dateUntil := FMDateTimeToDateTime(StrToFloat(sStop));  IDItem.Comments := sComment;  end;  IDItem.setListItem(clvSurrogates.Items[i]);  RefreshList;  SurrogateUpdated := fRawServerData <> ListViewToRaw;  end;  end;  end; |
| procedure clvSurrogatesDblClick | procedure | Ignites Add or Edit  procedure TfrmOptionsSurrogate.clvSurrogatesDblClick(Sender: TObject);  begin  inherited;  btnSurrEditClick(nil);  end; |
| procedure clvSurrogatesKeyDown | procedure | Implements K/b access to the action  procedure TfrmOptionsSurrogate.clvSurrogatesKeyDown(Sender: TObject;  var Key: Word; Shift: TShiftState);  begin  if Key = VK\_INSERT then  begin  if ssCtrl in Shift then  begin  btnSurrEditClick(nil);  Key := 0;  end;  end  else if Key = VK\_DELETE then  begin  if ssCtrl in Shift then  begin  btnRemoveClick(nil);  Key := 0;  end;  end;  inherited;  end; |
| procedure clvSurrogatesMouseDown | procedure | Ignites “Remove”  procedure TfrmOptionsSurrogate.clvSurrogatesMouseDown(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  begin  inherited;  if ssCtrl in Shift then  btnRemoveClick(nil);  end; |
| procedure FormCreate | procedure | Initializes the collection of the descriptors  procedure TfrmOptionsSurrogate.FormCreate(Sender: TObject);  begin  inherited;  fIDCollection := TCollection.Create(TIDItem);  {$IFDEF DEBUG\_AA}  pnlDebug.Visible := True;  {$ENDIF}  end; |
| procedure FormDestroy | procedure | Releases the descriptor collection  procedure TfrmOptionsSurrogate.FormDestroy(Sender: TObject);  begin  fIDCollection.Free;  inherited;  end; |

Class Methods (form TfrmOptionsSurrogate)

| Method Name | Procedure / Function | Description |
| --- | --- | --- |
| LoadServerData | procedure | Retrieves surrogate assignments from server.  procedure TfrmOptionsSurrogate.LoadServerData;  var  ts: TStringList;  begin  inherited;  fIDCollection.Clear;  clvSurrogates.Clear;  ts := TStringList.Create;  rpcGetSurrogateInfoList(ts);  try  if ts.Count > 1 then  begin  ts.Delete(0);  fRawServerData := ts.Text;  LoadListViewByStringList(ts);  fRawServerData := ListViewToRaw;  end;  finally  ts.Free;  end;  SurrogateUpdated := False;  end; |
| LoadListViewByStringList | procedure | Assigns clvSurrogates records based on TStringListData  procedure TfrmOptionsSurrogate.LoadListViewByStringList(aList: TStringList);  var  IDItem: TIDItem;  i: Integer;  sData, sName, sFrom, sUntil: String;  li: TListItem;  begin  if not assigned(aList) then  exit;  for i := 0 to aList.Count - 1 do  begin  ParseServerRecord(aList[i], sData, sName, sFrom, sUntil);  li := clvSurrogates.Items.Add;  IDItem := newTIDItem(fIDCollection, sData, '', sName, '',  FMDateTimeToDateTime(StrToFloat(sFrom)),  FMDateTimeToDateTime(StrToFloat(sUntil)));  IDItem.setListItem(li);  end;  end; |
| setRangeInfo | procedure | Calculates unassigned periods and adds corresponding records to the clvSurrogates  procedure TfrmOptionsSurrogate.setRangeInfo;  var  IDItem: TIDItem;  li: TListItem;  i: Integer;  dtLimit, dtNow, dtFrom, dtUntil, dtMax: TDateTime;  begin  inherited;  dtNow := Now;  dtLimit := dtNow + maxSurrPeriod;  dtMax := dtNow;  dtUntil := dtNow;  dtFrom := 0.0;  fIgnore := True;  i := 0;  while i < clvSurrogates.Items.Count do  begin  if assigned(clvSurrogates.Items[i].Data) then  begin  IDItem := TIDItem(clvSurrogates.Items[i].Data);  dtFrom := IDItem.dateFrom;  dtUntil := IDItem.dateUntil;  end  else  begin  {$IFDEF DEBUG\_AA}  MessageDlg('No Object assigned to the record <' + IntToStr(i) + '>!',  mtError, [mbOK], 0);  {$ENDIF}  continue;  end;  if dtUntil < dtNow then  begin  inc(i);  continue; // ignore ranges starting in the past  end;  if dtFrom < dtNow then  begin  dtMax := dtUntil;  inc(i);  continue;  end;  if IsOpen(i) then // no open records are expected while setting open ranges  begin  {$IFDEF DEBUG\_AA}  MessageDlg('No Open records expected at this time! (record: ' +  IntToStr(i) + ')', mtError, [mbOK], 0);  {$ENDIF}  inc(i);  continue;  end;  if (dtFrom - dtMax >= 2 \* dtGap) and ((dtFrom - dtMax) > 0) then  begin  li := clvSurrogates.Items.Insert(i);  IDItem := newTIDItem(fIDCollection, '', ciOpen, ciUnknownName, '',  dtMax + dtGap, dtFrom - dtGap);  IDItem.setListItem(li);  dtMax := dtUntil;  inc(i, 2);  end  else  begin  dtMax := dtUntil;  inc(i);  end;  end;  if dtLimit - dtUntil >= dtGap then  begin  li := clvSurrogates.Items.Insert(i);  IDItem := newTIDItem(fIDCollection, '', ciOpen, ciUnknownName, '',  dtUntil + dtGap, dtLimit);  IDItem.setListItem(li);  end;  fIgnore := False;  end; |
| clearRanges | procedure | Removes records from the clvSurrogates  procedure TfrmOptionsSurrogate.clearRanges;  var  i: Integer;  begin  i := 0;  while i < clvSurrogates.Items.Count do  begin  if IsOpen(i) then  clvSurrogates.Items.Delete(i)  else  inc(i);  end;  end; |
| mergeItems | procedure | Merges unassigned periods    procedure TfrmOptionsSurrogate.mergeItems;  var  i: Integer;  IDItem, IDPrev: TIDItem;  Begin  i := 1;  IDPrev := TIDItem(clvSurrogates.Items[0].Data);  while i < clvSurrogates.Items.Count do  begin  IDItem := TIDItem(clvSurrogates.Items[i].Data);  if IDItem.IDString = IDPrev.IDString then  begin  IDPrev.dateUntil := IDItem.dateUntil;  IDPrev.setListItem(clvSurrogates.Items[i - 1]);  clvSurrogates.Items.Delete(i);  end  else  begin  IDPrev := IDItem;  inc(i);  end;  end;  end; |
| reNumItems | procedure | Assigns numbers to the records of clvSurrogates  procedure TfrmOptionsSurrogate.reNumItems;  var  n, i: Integer;  begin  i := 0;  n := 1;  while i < clvSurrogates.Items.Count do  begin  if IsOpen(i) then  clvSurrogates.Items[i].Caption := ciOpen  else  begin  clvSurrogates.Items[i].Caption := IntToStr(n);  if (n = 1) and (i = 0) then  clvSurrogates.Items[i].SubItems[3] := ciActive;  inc(n);  end;  inc(i);  end;  end; |
| ParseServerRecord | procedure | Server record parser  procedure TfrmOptionsSurrogate.ParseServerRecord(aValue: String;  var sData, sName, sFrom, sUntil: String);  begin  sData := piece(aValue, U, 1);  sName := piece(aValue, U, 2);  sFrom := piece(aValue, U, 3);  sUntil := piece(aValue, U, 4);  end; |
| IsOpen | function | Returns TRUE if the selected period is unassigned  function TfrmOptionsSurrogate.IsOpen(anItem: Integer): Boolean;  begin  if assigned(clvSurrogates.Items[anItem].Data) then  Result := TIDItem(clvSurrogates.Items[anItem].Data)  .PublicName = ciUnknownName  else  Result := True;  end; |
| ListViewToRaw | function | Converts surrogate assignments to string for saving on server  function TfrmOptionsSurrogate.ListViewToRaw: String;  var  s: String;  IDItem: TIDItem;  SL: TStringList;  i: Integer;  begin  SL := TStringList.Create;  for i := 0 to clvSurrogates.Items.Count - 1 do  begin  if assigned(TIDItem(clvSurrogates.Items[i].Data)) then  begin  IDItem := TIDItem(clvSurrogates.Items[i].Data);  if IDItem.IsOpen then  continue;  s := IDItem.toRawString;  SL.Add(s);  end  {$IFDEF DEBUG\_AA}  else  ShowMessage('All Records should have an object assigned!');  {$ENDIF}  end;  Result := SL.Text;  SL.Free;  end; |
| setSurrogateUpdated | procedure | Setter of the SurrogateUpdated property  procedure TfrmOptionsSurrogate.setSurrogateUpdated(aValue: Boolean);  begin  fSurrogateUpdated := aValue;  stxtChanged.Visible := fSurrogateUpdated;  NotifyParent;  end; |
| setByCurrentItem | procedure | Assigns caption of btnSurrEdit based on selected record of the clvSurrogates list  procedure TfrmOptionsSurrogate.setByCurrentItem;  var  ind: Integer;  IDItem: TIDItem;  begin  ind := clvSurrogates.ItemIndex;  if ind < 0 then  begin  btnRemove.Enabled := False;  btnSurrEdit.Enabled := False;  end  else  begin  btnSurrEdit.Enabled := True;  IDItem := TIDItem(clvSurrogates.Items[clvSurrogates.ItemIndex].Data);  if assigned(IDItem) then  begin  btnRemove.Enabled := not IDItem.IsOpen;  if btnRemove.Enabled then  btnSurrEdit.Caption := '&Edit Surrogate'  else  btnSurrEdit.Caption := '&Add Surrogate';  end  else  ShowMessage('Item should be assigned!...');  end;  end; |
| ServerDeleteAll | function | Deletes currently assigned surrogates  function TfrmOptionsSurrogate.ServerDeleteAll: String;  var  s, sMsg: String;  bOK: Boolean;  SL: TStringList;  begin  SL := TStringList.Create;  SL.Text := fRawServerData;  while SL.Count > 0 do  begin  s := SL[0];  try  s := pieces(s, U, 1, 2);  rpcSetSurrogateInfo(s, bOK, sMsg);  except  On E: Exception do  begin  sMsg := E.Message;  bOK := False;  end;  end;  if not bOK then  begin  MessageDlg('Error deleting surrogate(s):' + CRLF + CRLF + sMsg, mtError,  [mbOK], 0);  Result := Result + sMsg + CRLF;  end;  SL.Delete(0);  end;  SL.Free;  end; |
| ServerSaveAll | function | Saves all records of the list view on server.  function TfrmOptionsSurrogate.ServerSaveAll: String;  var  sMsg: String;  bOK: Boolean;  SL: TStringList;  begin  Result := '';  SL := TStringList.Create;  SL.Text := ListViewToRaw;  while SL.Count > 0 do  begin  rpcSetSurrogateInfo(SL[0], bOK, sMsg);  if not bOK then  begin  MessageDlg('Error saving surrogate(s):' + CRLF + CRLF + sMsg, mtError,  [mbOK], 0);  Result := Result + sMsg + CRLF;  end;  SL.Delete(0);  end;  SL.Free;  end; |
| NotifyParent | procedure | Notifies parent form on changes to assignments  procedure TfrmOptionsSurrogate.NotifyParent;  var  obj: TObject;  begin  obj := Parent;  while (obj <> nil) and not(obj is TForm) do  obj := TWinControl(obj).Parent;  if obj <> nil then  SendMessage(TWinControl(obj).Handle, UM\_CHECKAPPLY, 0, 0);  end; |
| getItemID | function | Returns ID of assigned surrogate descriptor  function TfrmOptionsSurrogate.getItemID(anItem: Integer): String;  begin  if assigned(clvSurrogates.Items[anItem].Data) then  Result := TIDItem(clvSurrogates.Items[anItem].Data).IDString  else  Result := '';  end; |
| ApplyChanges | procedure | Saves changes to the server  function TfrmOptionsSurrogate.ApplyChanges: Boolean;  var  sMsg, sNewList, sRaw: String;  function getRawList(aString: String): String;  var  i: Integer;  lst: TStringList;  begin  Result := '';  lst := TStringList.Create;  rpcGetSurrogateInfoList(lst);  try  if lst.Count > 1 then  begin  for i := 1 to lst.Count - 1 do  Result := Result + piece(lst[i], U, 1) + U + piece(lst[i], U, 3) + U +  piece(lst[i], U, 4) + CRLF;  end;  finally  lst.Free;  end;  end;  begin  // check if the surrogates settings were updated since the last load  sRaw := getRawList(sNewList);  if fRawServerData <> sRaw then  begin  MessageDlg(msgSurrogateChangesSinceLastLoad  {$IFDEF DEBUG}  + CRLF + CRLF + 'Server before: ' + CRLF + fRawServerData + CRLF + CRLF +  'Server Now : ' + CRLF + sRaw  {$ENDIF}  , mtWarning, [mbOK], 0);  Result := False;  end  else  begin  Result := not SurrogateUpdated;  if Result then // no need to save changes (or update the table)  exit;  sMsg := ServerDeleteAll;  if sMsg = '' then  sMsg := ServerSaveAll;  Result := sMsg = '';  if Result then  SurrogateUpdated := False  else  MessageDlg('Error applying saving surrogates changes' + CRLF + CRLF +  sMsg, mtError, [mbOK], 0);  end;  UpdateWithServerData;  end; |
| UpdateWithServerData | procedure | Updates table with data from server  procedure TfrmOptionsSurrogate.UpdateWithServerData;  begin  LoadServerData;  RefreshList;  end; |
| RefreshList | procedure | Updates screen  procedure TfrmOptionsSurrogate.RefreshList;  begin  clearRanges;  setRangeInfo;  mergeItems;  reNumItems;  if clvSurrogates.Items.Count = 1 then  clvSurrogates.ItemIndex := 0;  setByCurrentItem;  end; |

Special References

No changes

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Fields (form TfrmOptionsSurrogate)

| Class Fields Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| fIDCollection | TCollection | private | Collection of surrogate descriptors |
| fRawList | String | Private | Surrogates assignment on Server |
| fIgnore | Boolean | Private | Flag preventing recurrent calls of clvSurrogatesChange |
| fSurrogateUpdated | Boolean | Private | Property SurrogateUpdated |

Class Properties (form TfrmOptionsSurrogate)

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| SurrogateUpdated | Boolean | Public | Indicator of changes to the surrogate assignments |

Unit Variables

| Name | Type | Description |
| --- | --- | --- |
| frmOptionsSurrogate | TfrmOptionsSurrogate | Default variable of type TfrmOptionsSurrogate |

Unit Constants

| Name | Type | Description |
| --- | --- | --- |
| fmtListDateTime = 'mm/dd/yyyy@hh:nn'; | String | Date/ time format for surrogate presentation |
| fmtListDateTimeSec = 'mm/dd/yyyy@hh:nn:ss' | String | Debug version to control second values |
| ciOpen = '...'; | String | Used instead on the record number for presentation of the unassigned records |
| ciUnknownName = '.....'; | String | Used instead on the surrogate name for presentation of the unassigned records |
| ciActive = 'Active'; | String | Active surrogate indicator |
| maxSurrPeriod = 365.25 \* 4; | float | Latest date available for assignment (4 years from the current date) |
| dtGap = 0.0; | float | Required Gap between assignment of the surrogates |
| msgSurrogateRemove = 'Remove Surrogate?'; | String | Removal confirmation text |
| msgSurrogateChangesSinceLastLoad =  'Surrogate settings were changed since the last request' + CRLF +  'Please review the current settings prior to saving the updates'; | String | Informational message for cases the surrogate list was changed since the last call to the server. (Cases the list was updated from roll and scroll) |

Unit Methods

| Name | Procedure / Function | Description |
| --- | --- | --- |
| StrDateToDate | function | Converts string date to TDateTime  function StrDateToDate(aDate: String): TDateTime;  var  dtDate, dtTime: Real;  sDate, sTime: String;  begin  // expected format: 'mm/dd/yyyy@hh:nn';  sDate := piece(aDate, '@', 1);  sTime := piece(aDate, '@', 2);  dtDate := strToDate(sDate);  dtTime := strToTime(sTime);  Result := dtDate + dtTime;  end; |
| StrDateToFMDate | function | Converts String date To FM Date time  function StrDateToFMDate(aDate: String): TDateTime;  begin  try  Result := DateTimeToFMDateTime(StrDateToDate(aDate));  except  on E: Exception do  begin  Result := 0;  {$IFDEF DEBUG}  MessageDlg('Error converting string to FM date:' + CRLF + CRLF +  E.Message, mtError, [mbOK], 0);  {$ENDIF}  end;  end;  end; |
| strDateToFMDateStr | function | Converts String date to FM date string  function StrDateToFMDateStr(aDate: String): String;  begin  Result := FloatToStr(StrDateToFMDate(aDate));  end; |
| function newTIDItem | function | Returns new TIdItem  function newTIDItem(aCollection: TCollection;  anID, aCaption, aName, aComment: String; aFrom, aUntil: TDateTime): TIDItem;  begin  Result := TIDItem.Create(aCollection);  Result.IDString := anID;  Result.Caption := aCaption;  Result.PublicName := aName;  Result.Comments := aComment;  Result.dateFrom := aFrom;  Result.dateUntil := aUntil;  end; |

Additional classes defined in the unit

| Name | Accessor Type | Description |
| --- | --- | --- |
| TIDItem | TCollectionItem | Class items represent surrogate records.  TIDItem = class(TCollectionItem)  private  public  IDString: String;  Caption, PublicName: String;  Comments: String;  dateFrom, dateUntil: TDateTime;  function strFrom: String;  function strUntil: String;  function toRawString: String;  procedure setListItem(anItem: TListItem);  function getInfo: String;  function getInfoDebug: String;  function IsOpen: Boolean;  end; |

Class TIdItem Fields

| Class Fields Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| IDString | String | public | String presentation of the surrogate item |
| Caption | String | Public | Used as Caption for surrogate presentation |
| PublicName | String | Public | Name to use for presentation |
| Comments | String | public | Used as comment for surrogate descriptor |
| dateFrom | TDateTime | public | Surrogate assignment date |
| dateUntil | TDateTime | public | Surrogate assignment date |

Class TIdItem Methods

| Method Name | Procedure / Function | Description |
| --- | --- | --- |
| strFrom | Function | String presentation of the “From” date  function TIDItem.strFrom: String;  begin  Result := FormatDateTime(fmtListDateTime, dateFrom);  end; |
| strUntil | Function | String presentation of the “Until” date  function TIDItem.strUntil: String;  begin  Result := FormatDateTime(fmtListDateTime, dateUntil);  end; |
| toRawString | Function | String presentation of the descriptor for filing to server  function TIDItem.toRawString: String;  begin  Result := IDString + U + FloatToStr(DateTimeToFMDateTime(dateFrom)) + U +  FloatToStr(DateTimeToFMDateTime(dateUntil));  end |
| setListItem | procedure | Assignment of the presentation Item according to the descriptor  procedure TIDItem.setListItem(anItem: TListItem);  begin  if assigned(anItem) then  begin  while anItem.SubItems.Count < 4 do  anItem.SubItems.Add('');  anItem.Caption := Caption;  anItem.SubItems[0] := PublicName;  anItem.SubItems[1] := FormatDateTime(fmtListDateTime, dateFrom);  anItem.SubItems[2] := FormatDateTime(fmtListDateTime, dateUntil);  anItem.SubItems[3] := Comments;  anItem.Data := self;  end;  end; |
| getInfo | Function | Returns information on descriptor for user review  function TIDItem.getInfo: String;  begin  Result := PublicName + ' ' + FormatDateTime(fmtListDateTime, dateFrom) +  ' .. ' + FormatDateTime(fmtListDateTime, dateUntil);  end; |
| getInfoDebug | function | Returns descriptor in debug format  function TIDItem.getInfoDebug: String;  begin  Result := IDString + ' ' + Caption + ' ' + PublicName + ' ' +  FormatDateTime(fmtListDateTimeDebug, dateFrom) + ' .. ' +  FormatDateTime(fmtListDateTimeDebug, dateUntil) + ' ' + Comments;  end; |
| IsOpen | function | Returns “True” for unassigned periods  function TIDItem.IsOpen: Boolean;  begin  Result := PublicName = ciUnknownName;  end; |

File fSurrogateEdit.pas changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| TfrmSurrogateEdit | File fSurrogateEdit.pas  Implements TfrmSurrogate dialog | | | |
| Description | Implementation of class TfrmSurrogate – selector of the surrogate parameters | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| Form layout | The dialog allows user select the name of the surrogate and the date range of assignment. | | | |
| **Class TSurrogateEdit Modified Logic (New code added)** | | | | |
| See files fSurrogateEdit.pas; fSurrogateEdit.dfm | | | | |

Components on Form

| Name | Type | Description |
| --- | --- | --- |
| VA508StaticTextName | TVA508StaticText | Label of the cboSurrogates combo box |
| stxtChanged | TStaticText | Indicator of changes. Component is visible when the user updates surrogate settings |
| stxtRangeInfo | TVA508StaticText | Date Range available for selection. Min and Max values. |
| cboSurrogate | TORComboBox | Selector of surrogate name |
| VA508StaticText1 | TVA508StaticText | Label of ordtbStart date box component |
| VA508StaticText2 | TVA508StaticText | Label of ordtbStop date box component |
| ordtbStart | TORDateBox | Selector of date/time to use as the beginning of the surrogate assignment. |
| ordtbStop | TORDateBox | Selector of date/time to use as the end of the surrogate assignment. |
| btnRestore | TButton | Restore the original values of the date range |
| btnOK | TButton | Button closes dialog with result mrOK |
| btnCacnel | TButton | Button closing dialog with the result of mrCancel |
| pnlSurrogateTools | TPanel | Container of the dialog components |
| pnlBottom | TPanel | Container of the dialog buttons |

Events

| Name | Type | Description |
| --- | --- | --- |
| FormShow | OnShow | Puts focus on cboSurrogate |
| FormCloseQuery | OnCloseQuery | Validates if the name was assigned |
| btnRestoreClick | OnClick | Restores the original values of settings |
| btnCancelClick | OnClick | Setting flag fClose to control validation of parameters |
| FormKeyDown | OnKeyDown | Verifies if the VK\_ESC was pressd and the form should be closed |
| ordtbStartDateDialogClosed | OnDateDialogClosed | Updates indicator of changes on closing of the date/time selector |
| ordtbStopDateDialogClosed | OnDateDialogClosed | Updates indicator of changes on closing of the date/time selector |
| cboSurrogateChange | OnChange | Sets status of the buttons |
| cboSurrogateNeedData | OnNeedData | Look ups for the surrogate on server. |

Class Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| SetValues | procedure | Assigns values to form components. |
| setButtonStatus | procedure | Validates status of the OK button |

Special References

No changes

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Events

No changes

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Fields

| Class Fields Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| fData | Integer | Private | Surrogates assignment on Server |
| fStart | String | Private | Beginning of the selected date range |
| fStop | String | Private | End of the selected date range |
| fMin | String | Private | Min value allowed for selection of date/time |
| fMax | String | Private | Max value allowed for selection of date/time |
| fSurrogate | String | Private | Initial value of the surrogate name |
| fCancel | Boolean | Private | Flag controlling validation of input. (If set to TRUE no validation is needed) |

Class Properties

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Unit Variables

| Name | Type | Description |
| --- | --- | --- |
| frmSurrogateEdit | TfrmSurrogateEdit | Default variable of type TfrmSurrogateEdit |

Unit Methods

| Name | Procedure / Function | Description |
| --- | --- | --- |
| EditSurrogate | function | Allows user select surrogate settings |

File uConst.pas changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Constant added** | **UM\_CHECKAPPLY = (WM\_USER + 9600);**  Constant is used to identify message sent by TfrmOptionSurrogate form to notify parent form on changes in surrogates assignment | | | |

Components on Form

N/A

Events

N/A

Special References

N/A

Class Events

N/A

Unit Methods

N/A

Class Properties

N/A

Uses Clause

No changes

File rOptions.pas changes

| Unit | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| rOptions | File rOptions.pas  Modifications to implement Multiple surrogates management functionality (Saving/restoring surrogates settings to/from server) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Methods Added** | **procedure rpcGetSurrogateInfoList(const aResult:TStrings)** | | | |
| **Unit rOptions Modified Logic (New code added)** | | | | |
| **// NSR20071216 AA 2015/11/30 --------------------------------------------- begin**  **// NSR20071216 mnj- RPC 'ORWTPP GETSURRS' addded to return multiple Surrogates.**  **procedure rpcGetSurrogateInfoList(const aResult:TStrings);**  **begin**  **tCallV(aResult,'ORWTPP GETSURRS', [nil]);**  **MixedCaseList(aResult);**  **end;**  **// NSR20071216 AA 2015/11/30 ----------------------------------------------- end** | | | | |
| **Comments** | New code is shown in **bold** | | | |

File ORDtTmCal.pas changes

| Unit | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| ORDtTm | File ORDtTmCal.pas  Modifications to implement range check in TORCalendar class | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Comments** | Unit modified to enhance CPRSLib components with support of range check. | | | |
| **Unit Methods Added** |  | | | |
| **Comments** | New code is shown in **bold** | | | |

Uses Clause changes

Reference on System.UITypes, System.DateUtils, ORDtTm added to implementation section

Components Modified

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | **TDateRange = class(TObject)** | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | Object provides properties to define date range and validate dates against rangew boundaries | | | |
| **Fields** | **fMinDate: Double** | | | |
| **fMaxDate: Double** | | | |
| **Properties** | **MinDate: Double** | | | |
| **MaxDate: Double** | | | |
| **Methods Added/Modified** | New | **procedure TDateRange.SetMinDate(Const aMinDte: Double);**  **begin**  **if (fMaxDate <> -1) and (aMinDte > fMaxDate) then**  **begin**  **raise Exception.Create('Min date cannot be after Max date');**  **Exit;**  **end;**  **fMinDate := aMinDte;**  **end;** | | |
|  | New | **procedure TDateRange.SetMaxDate(const aMaxDte: Double);**  **begin**  **if (fMinDate <> -1) and (aMaxDte < fMinDate) then**  **begin**  **raise Exception.Create('Max date cannot be before Min date');**  **Exit;**  **end;**  **fMaxDate := aMaxDte;**  **end;** | | |
|  | New | **function TDateRange.IsBetweenMinAndMax(const LookupDate: TDateTime): Boolean;**  **var**  **dt: TDateTime;**  **begin**  **dt := DateOf(LookupDate);**  **Result := (dt >= DateOf(fMinDate)) and (dt <= DateOf(fMaxDate));**  **end;** | | |
|  | New | **function TDateRange.IsFullDay(const aDate: TDateTime): Boolean;**  **var**  **\_min, \_max, dtMin, dtMax: TDateTime;**  **begin**  **\_min := TDateTime(MinDate);**  **\_max := TDateTime(MaxDate);**  **dtMin := round(aDate) + 1 / (24 \* 60 \* 60);**  **dtMax := round(aDate + 1) - 1 / (24 \* 60 \* 60);**  **Result := (\_min < dtMin) and (dtMax < \_max);**  **end;** | | |
| **Comments** | New code is shown in **bold** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TORCalendar = class(ORDtTmCal.TORCalendar) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | Enhances standard calendar with the range checking functionality | | | |
| **Fields** | New | **fValidRange: TDateRange** | | |
| **Properties** | New | **property ValidRange: TDateRange read fValidRange write SetValidRange;** | | |
| **Methods** | New | **procedure TORCalendar.SetValidRange(aRange: TDateRange);**  **begin**  **fValidRange := aRange;**  **if ValidRange.IsBetweenMinAndMax(ServerToday) then**  **CalendarDate := ServerToday**  **else**  **CalendarDate := ValidRange.MinDate;**  **end;** | | |
|  | New | **constructor TListBox.Create(AOwner: TComponent);**  **begin**  **inherited Create(AOwner);**  **fMinTime := -1;**  **fMaxTime := -1;**  **end;** | | |
|  | Modified | procedure TORCalendar.DrawCell(ACol, ARow: Longint; ARect: TRect;  AState: TGridDrawState); | | |
|  | New | **function TORCalendar.getMinDateTime: TDateTime;**  **begin**  **if assigned(ValidRange) then**  **Result := ValidRange.MinDate**  **else**  **Result := -1.0;**  **end;** | | |
|  | New | **function TORCalendar.getMaxDateTime: TDateTime;**  **begin**  **if assigned(ValidRange) then**  **Result := ValidRange.MaxDate**  **else**  **Result := -1.0;**  **end;** | | |
|  | Modified | function TORCalendar.IsFullDay(aDate: TDateTime): Boolean;  begin  Result := False;  **if assigned(ValidRange) then**  **Result := ValidRange.IsFullDay(aDate);**  end; | | |
|  | Modified | function TORCalendar.IsBetweenMinAndMax(const LookupDate: TDateTime): Boolean;  begin  Result := False;  **if assigned(ValidRange) then**  **Result := ValidRange.IsBetweenMinAndMax(LookupDate);**  end; | | |
|  | Modified | function TORCalendar.SelectCell(ACol, ARow: Longint): Boolean;  var  TheText: string;  DteToChk: TDateTime;  \_min, \_max: TDateTime;  begin  Result := (Inherited);  TheText := CellText[ACol, ARow];  if TheText = '' then  Exit;  DteToChk := EncodeDate(Year, Month, StrToIntDef(TheText, 0));  **if assigned(ValidRange) then**  **begin**  **if Result then**  **begin**  **\_min := getMinDateTime;**  **\_max := getMaxDateTime;**  **if \_min <> -1 then**  **Result := DateOf(DteToChk) >= DateOf(FloatToDateTime(\_min));**  **if \_max <> -1 then**  **Result := Result and**  **(DateOf(DteToChk) <= DateOf(FloatToDateTime(\_max)));**  **end;**  **end;**  end; | | |
|  | Modified | procedure TORCalendar.DrawCell(ACol, ARow: Longint; ARect: TRect;  AState: TGridDrawState);  var  TheText: string;  DteToChk: TDateTime;  CurMonth, CurYear, CurDay: Word;  UseColor: TColor;  \_min, \_max: TDateTime;  begin  TheText := CellText[ACol, ARow];  with ARect, Canvas do  begin  **if assigned(ValidRange) then**  begin  \_min := getMinDateTime;  \_max := getMaxDateTime; | | |
| **Comments** | New code is shown in **bold** | | | |

File ORDtTm.pas changes

| Unit | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| ORDtTm | File ORDtTm.pas  Modifications to implement Multiple surrogates management functionality (Saving/restoring surrogates settings to/from server) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Comments** | Unit modified to enhance CPRSLib components with support of range check. | | | |
| **Methods Added** | **function stripCharSet(AString: String; aSet: TSysCharSet): String;**  **var**  **i: integer;**  **begin**  **Result := AString;**  **for i := Length(Result) downto 1 do**  **if not CharInSet(Result[i], aSet) then**  **Delete(Result, i, 1);**  **end;** | | | |
| **function stripChars(AString: String): String;**  **begin**  **Result := stripCharSet(AString, ['0' .. '9']);**  **end;** | | | |
| **Comments** | New code is shown in **bold** | | | |

Uses Clause changes

Reference on System.UITypes added to implementation section

Components modified

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TListBox = class(StdCtrls.TListBox) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | Standard class TListBox updated to use with the time selector of the TORDateTimeDlg. New fields added to keep values of min and max times allowed for selection in the dialog. Procedures drawing the items on the screen updated to use different colors for available and disabled time items. | | | |
| **Fields** | New | **fMinTime: Integer** | | |
| New | **fMaxTime: Integer** | | |
| **Properties** | New | **property MaxTime: integer read fMaxTime write fMaxTime;** | | |
| New | **property MinTime: integer read fMinTime write fMinTime;** | | |
|  | New | **procedure TListBox.SetItemIndex(const Value: Integer);**  **var**  **TempStr: String;**  **begin**  **// Check if this is valid or not**  **if Value <> -1 then**  **begin**  **TempStr := self.Items[Value];**  **TempStr := stripChars(TempStr);**  **if fMinTime > -1 then**  **if StrToIntDef(TempStr, -1) < fMinTime then**  **exit;**  **if fMaxTime > -1 then**  **if StrToIntDef(TempStr, -1) > fMaxTime then**  **exit;**  **end;**  **inherited;**  **end** | | |
| **Methods** | New | **procedure TListBox.KeyDown(var Key: Word; Shift: TShiftState);**  **var**  **ind: Integer;**  **TempStr: String;**  **begin**  **if Key in [VK\_UP, VK\_DOWN, VK\_LEFT, VK\_RIGHT, VK\_PRIOR, VK\_NEXT] then**  **begin**  **case Key of**  **VK\_UP, VK\_LEFT:**  **ind := ItemIndex - 1;**  **VK\_DOWN, VK\_RIGHT:**  **ind := ItemIndex + 1;**  **VK\_PRIOR:**  **ind := ItemIndex - 11;**  **VK\_NEXT:**  **ind := ItemIndex + 11;**  **else**  **ind := ItemIndex;**  **end;**  **if (ind < 0) or (ind >= Items.Count) then**  **Key := 0**  **else**  **begin**  **TempStr := self.Items[ind];**  **TempStr := stripChars(TempStr);**  **if fMinTime > -1 then**  **if StrToIntDef(TempStr, -1) < fMinTime then**  **Key := 0;**  **if fMaxTime > -1 then**  **if StrToIntDef(TempStr, -1) > fMaxTime then**  **Key := 0;**  **end;**  **end;** | | |
| New | **procedure TListBox.CNDrawItem(var Message: TWMDrawItem);**  **var**  **State: TOwnerDrawState;**  **TempStr: String;**  **begin**  **inherited;**  **if (fMinTime > -1) or (fMaxTime > -1) then**  **begin**  **with Message.DrawItemStruct{$IFNDEF CLR}^{$ENDIF} do**  **begin**  **TempStr := self.Items[ItemID];**  **TempStr := stripChars(TempStr);**  **State := TOwnerDrawState(LoWord(itemState));**  **if fMinTime > -1 then**  **begin**  **if StrToIntDef(TempStr, -1) < fMinTime then**  **begin**  **if (odFocused in State) then**  **DrawFocusRect(hDC, rcItem);**  **end;**  **end**  **else if fMaxTime > -1 then**  **begin**  **if StrToIntDef(TempStr, -1) > fMaxTime then**  **begin**  **if (odFocused in State) then**  **DrawFocusRect(hDC, rcItem);**  **end;**  **end;**  **end;**  **end;**  **end;** | | |
| New | **procedure TListBox.DrawItem(Index: Integer; Rect: TRect;**  **State: TOwnerDrawState);**  **var**  **FontClr, BrshClr: TColor;**  **TempStr: String;**  **begin**  **// Draw invalid here**  **TempStr := self.Items[Index];**  **TempStr := stripChars(TempStr);**  **With self.Canvas do**  **begin**  **BrshClr := clWindow;**  **FontClr := clWindowText;**  **if (odSelected in State) then**  **begin**  **BrshClr := clHighlight;**  **FontClr := clHighlightText;**  **end;**  **if fMinTime > -1 then**  **begin**  **if StrToIntDef(TempStr, -1) < fMinTime then**  **BrshClr := clLtGray;**  **end;**  **if fMaxTime > -1 then**  **begin**  **if StrToIntDef(TempStr, -1) > fMaxTime then**  **BrshClr := clLtGray;**  **end;**  **Brush.Color := BrshClr;**  **Font.Color := FontClr;**  **FillRect(Rect);**  **TextRect(Rect, Rect.Left, Rect.Top, Items.Strings[Index]);**  **end;**  **end;** | | |
| New | **constructor TListBox.Create(AOwner: TComponent);**  **begin**  **inherited Create(AOwner);**  **fMinTime := -1;**  **fMaxTime := -1;**  **end;** | | |
| New | **destructor TListBox.Destroy;**  **begin**  **inherited;**  **end;** | | |
|  |  |  | | |
| **Comments** | New code is shown in **bold** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TORfrmDtTm | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** |  | | | |
| **Fields** |  |  | | |
| **Properties** |  |  | | |
| **Methods** | New | **procedure TORfrmDtTm.FormKeyDown(Sender: TObject; var Key: Word;**  **Shift: TShiftState);**  **begin**  **if Key = VK\_ESCAPE then**  **ModalResult := mrCancel;**  **end;** | | |
|  | New | **procedure TORfrmDtTm.setTimeListsBySelectedDate;**  **begin**  **lstHour.MinTime := -1;**  **lstHour.MaxTime := -1;**  **lstMinute.MinTime := -1;**  **lstMinute.MaxTime := -1;**  **// Check for Min and Max of calendar**  **if DateOf(calSelect.CalendarDate) = DateOf(calSelect.MinDateTime) then**  **begin**  **lstHour.MinTime := HourOf(calSelect.MinDateTime);**  **lstMinute.MinTime := MinuteOf(calSelect.MinDateTime);**  **if TimeOf(calSelect.CalendarDate) < calSelect.MinDateTime then**  **begin**  **lstHour.ItemIndex := HourOf(calSelect.MinDateTime);**  **lstMinute.ItemIndex := MinuteOf(calSelect.MinDateTime);**  **end;**  **end**  **else if DateOf(calSelect.CalendarDate) = DateOf(calSelect.MaxDateTime) then**  **begin**  **lstHour.MaxTime := HourOf(calSelect.MaxDateTime);**  **lstMinute.MaxTime := MinuteOf(calSelect.MaxDateTime);**  **if TimeOf(calSelect.CalendarDate) > calSelect.MaxDateTime then**  **begin**  **lstHour.ItemIndex := HourOf(calSelect.MaxDateTime);**  **lstMinute.ItemIndex := MinuteOf(calSelect.MinDateTime);**  **end;**  **end**  **else // Date within range**  **begin**  **lstHour.ItemIndex := 6;**  **lstMinute.ItemIndex := 0;**  **end;**  **lstHour.Repaint;**  **lstMinute.Repaint;**  **end;** | | |
|  | New | **function TORfrmDtTm.IsRangeMode: Boolean;**  **begin**  **Result := (calSelect.fValidRange.MaxDate <> -1) or**  **(calSelect.fValidRange.MinDate <> -1)**  **end;** | | |
|  | Modified | procedure TORfrmDtTm.calSelectChange(Sender: TObject);  **procedure setTimeText;**  **var**  **s: String;**  **begin**  **s := ':';**  **if (lstHour.MinTime <> -1) then**  **s := stripChars(lstHour.Items[lstHour.ItemIndex]) + s;**  **if (lstMinute.MinTime <> -1) then**  **s := s + stripChars(lstMinute.Items[lstMinute.ItemIndex]);**  **if s = ':' then**  **s := '';**  **FFromSelf := True;**  **txtTime.Text := s;**  **FFromSelf := False;**  **end;**  begin  lblDate.Caption := FormatDateTime('mmmm d, yyyy', calSelect.CalendarDate);  FNowPressed := False;  if ScreenReaderSystemActive then  begin  // TxtDateSelected.Caption := lblDate.Caption;  TxtDateSelected.Caption := Label1.Caption + ' ' + lblDate.Caption;  GetScreenReader.Speak(lblDate.Caption);  end;  **// check if the valid range was set**  **if IsRangeMode then**  **begin**  **// Now need to enforce invalid times**  **setTimeListsBySelectedDate;**  **setTimeText;**  **setButtonStatus;**  **end;**  end; | | |
|  | Modified | procedure TORfrmDtTm.cmdTodayClick(Sender: TObject);  begin  calSelect.CalendarDate := ServerToday;  **// cmdToday is disabled if out of assigned valid range**  **if not IsRangeMode then**  **begin**  **lstHour.ItemIndex := -1;**  **lstMinute.ItemIndex := -1;**  **txtTime.Text := '';**  **end;**  end; | | |
|  | New | **procedure TORfrmDtTm.setButtonStatus;**  **var**  **dt: TDateTime;**  **begin**  **if IsRangeMode then**  **begin**  **cmdNow.Enabled := calSelect.fValidRange.IsBetweenMinAndMax(Now);**  **cmdToday.Enabled := calSelect.fValidRange.IsBetweenMinAndMax(ServerToday);**  **dt := calSelect.CalendarDate;**  **cmdMidnight.Enabled := calSelect.fValidRange.IsBetweenMinAndMax**  **(round(dt) + 1 - 1 / (24 \* 60));**  **end;**  **end;** | | |
|  | New | **procedure TORfrmDtTm.setRangeTimeBoundaries;**  **var**  **HourSel, TempStr: String;**  **begin**  **// only in case the valid range is assigned**  **if (calSelect.fValidRange.MaxDate = -1) and**  **(calSelect.fValidRange.MinDate = -1) then**  **exit;**  **// Need to set the min minute**  **lstMinute.MinTime := -1;**  **lstMinute.MaxTime := -1;**  **// Grab the hour selected**  **if lstHour.ItemIndex < 0 then**  **HourSel := '0'**  **else**  **HourSel := stripChars(lstHour.Items[lstHour.ItemIndex]);**  **// Now need to enforce invalid times**  **if DateOf(calSelect.CalendarDate) = DateOf(calSelect.MinDateTime) then**  **begin // date is on the range boundary**  **if StrToIntDef(HourSel, -1) <= HourOf(calSelect.MinDateTime) then**  **begin // reset minutes of the boundary hour**  **lstHour.ItemIndex := HourOf(calSelect.MinDateTime);**  **lstMinute.MinTime := MinuteOf(calSelect.MinDateTime);**  **// calculate min index**  **if (lstMinute.ItemIndex = -1) // minutes were not selected**  **or (lstMinute.ItemIndex < MinuteOf(calSelect.MinDateTime)) then**  **// assign min index if mot assigned**  **lstMinute.ItemIndex := MinuteOf(calSelect.MinDateTime)**  **end**  **else**  **// reset minutes**  **lstMinute.ItemIndex := 0; // -1;**  **end**  **else if DateOf(calSelect.CalendarDate) = DateOf(calSelect.MaxDateTime) then**  **begin // date is on the range boundary**  **if StrToIntDef(HourSel, -1) >= HourOf(calSelect.MaxDateTime) then**  **begin**  **lstHour.ItemIndex := HourOf(calSelect.MaxDateTime);**  **lstMinute.MaxTime := MinuteOf(calSelect.MaxDateTime);**  **// calculate min index**  **if (lstMinute.ItemIndex = -1) or**  **(lstMinute.ItemIndex > MinuteOf(calSelect.MinDateTime)) then**  **// assign min index if mot assigned**  **lstMinute.MaxTime := MinuteOf(calSelect.MaxDateTime);**  **end**  **else**  **// reset minutes**  **lstMinute.ItemIndex := 0; // was -1;**  **end;**  **lstMinute.Repaint;**  **end;** | | |
|  | Modified | procedure TORfrmDtTm.lstHourClick(Sender: TObject);  begin  **setRangeTimeBoundaries; // NSR20071216 AA 2016-01-29**  if lstHour.ItemIndex = 0 then  lstMinute.Items[0] := ':01 --'  else  lstMinute.Items[0] := ':00 --'; // <------ NEW CODE  lstMinuteClick(self);  end; | | |
|  | New | **procedure TORfrmDtTm.lstHourEnter(Sender: TObject);**  **begin**  **setRangeTimeBoundaries; // NSR20071216 AA 2016-01-29**  **end;** | | |
|  | New | **function TORfrmDtTm.TimeIsValid: Boolean;**  **var**  **tmpHour, TmpMinute: Integer;**  **begin**  **if not TimeIsRequired then**  **Result := True**  **else**  **begin**  **Result := False;**  **if Length(txtTime.Text) > 0 then**  **begin**  **// Check for invalide hour**  **tmpHour := StrToIntDef(Piece(Trim(txtTime.Text), ':', 1), -1);**  **if ((lstHour.MinTime > -1) and (tmpHour < lstHour.MinTime) or**  **(lstHour.MaxTime > -1) and (tmpHour > lstHour.MaxTime)) then**  **begin**  **txtTime.Text := '';**  **InfoBox('Please enter a valid time.', 'Invalid Time', MB\_OK);**  **// exit;**  **end**  **else**  **begin**  **// Check for invalide minute**  **TmpMinute := StrToIntDef(Piece(Trim(txtTime.Text), ':', 2), -1);**  **if ((lstMinute.MinTime > -1) and (TmpMinute < lstMinute.MinTime) or**  **(lstMinute.MaxTime > -1) and (TmpMinute > lstMinute.MaxTime)) then**  **begin**  **txtTime.Text := '';**  **InfoBox('Please enter a valid time.', 'Invalid Time', MB\_OK);**  **// exit;**  **end**  **else**  **Result := True;**  **end;**  **end;**  **end;**  **end;** | | |
|  | Modified | procedure TORfrmDtTm.cmdOKClick(Sender: TObject);  var  X: string;  tmpHour, TmpMinute: Integer;  begin  if TimeIsRequired and (Length(txtTime.Text) = 0) then  begin  InfoBox('An entry for time is required.', 'Missing Time', MB\_OK);  exit;  end;  if Length(txtTime.Text) > 0 then  begin  X := Trim(txtTime.Text);  if (X = '00:00') or (X = '0:00') or (X = '00:00:00') or (X = '0:00:00') then  X := '00:01'; // <------- CHANGED CODE  StrToTime(X);  txtTime.Text := X;  end;  **if TimeIsValid then // NSR20071216 AA 2016-01-29**  ModalResult := mrOK;  end; | | |
| **Comments** |  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TORDateTimeDlg | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | TORDateTimeDlg class modified to implement range checking. New property added to keep the range allowed for selection of the date and time. Class methods modified to add validation of the selected date. | | | |
| **Fields** | New | **fDateRange: TDateRange** | | |
| **Properties** | New | **property DateRange: TDateRange read fDateRange write setDateRange;** | | |
| New | **property DateTime: TDateTime read FDateTime write FDateTime;** | | |
| **Methods** | New | **procedure TORDateTimeDlg.setDateRange(aRange: TDateRange);**  **begin**  **fDateRange := aRange;**  **end;** | | |
|  | Modified | function TORDateTimeDlg.Execute: Boolean;  const  HORZ\_SPACING = 8;  var  frmDtTm: TORfrmDtTm;  **procedure setRange;**  **begin**  **with frmDtTm do**  **begin**  **if Assigned(self.DateRange) then**  **begin**  **if (self.DateRange.fMinDate <> -1) or (self.DateRange.fMaxDate <> -1)**  **then**  **Caption := Caption + ' between ' + FormatDateTime('mmm dd,YY@hh:nn ',**  **self.DateRange.fMinDate) + ' and ' +**  **FormatDateTime('mmm dd,YY@hh:nn', self.DateRange.fMaxDate);**  **calSelect.ValidRange := self.DateRange;**  **calSelect.CalendarDate := self.DateTime;**  **setButtonStatus;**  **end;**  **end;**  **end;**  begin  frmDtTm := TORfrmDtTm.Create(Application);  try  with frmDtTm do  begin  **setRange; // NSR20071216 AA 2016-01-22**  if Frac(FDateTime) > 0  // then txtTime.Text := FormatDateTime('h:nn ampm', FDateTime); // if ampm time  then  txtTime.Text := FormatDateTime('hh:nn', FDateTime); // if military time  if RequireTime then  TimeIsRequired := True;  if DateOnly then  begin  txtTime.Visible := False;  lstHour.Visible := False;  lstMinute.Visible := False;  cmdNow.Visible := False;  cmdMidnight.Visible := False;  bvlFrame.Width := bvlFrame.Width - txtTime.Width - HORZ\_SPACING;  cmdOK.Left := cmdOK.Left - txtTime.Width - HORZ\_SPACING;  cmdCancel.Left := cmdOK.Left;  ClientWidth := ClientWidth - txtTime.Width - HORZ\_SPACING;  end;  Result := (ShowModal = IDOK);  if Result then  begin  FDateTime := Int(calSelect.CalendarDate);  if Length(txtTime.Text) > 0 then  FDateTime := FDateTime + StrToTime(txtTime.Text);  if FNowPressed then  FRelativeTime := 'NOW'  else  FRelativeTime := RelativeDateTime(FDateTime);  end;  end;  finally  frmDtTm.Free;  end;  end; | | |
| **Comments** | New code is shown in **bold** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TORDateBox | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | Class modified to implement range checking. New property added to keep value of allowed range. | | | |
| **Fields** | New | **fDateRange: TDateRange** | | |
|  | New | **fDateSelected: TDateTime** | | |
| **Properties** | New | **property DateSelected: TDateTime read getDateSelected write setDateSelected;** | | |
| New | **property DateRange: TDateRange read fDateRange write fDateRange;** | | |
| **Methods** | New | **destructor TORDateBox.Destroy();**  **begin**  **inherited;**  **fDateRange.Free;**  **end;** | | |
|  | New | **function TORDateBox.getDateSelected: TDateTime;**  **begin**  **Result := fDateSelected;**  **end;** | | |
|  | New | **procedure TORDateBox.setDateSelected(aDate: TDateTime);**  **begin**  **fDateSelected := aDate;**  **end;** | | |
|  | Modified | constructor TORDateBox.Create(AOwner: TComponent);  begin  inherited Create(AOwner);  FButton := TORDateButton.Create(self);  FButton.Parent := self;  FButton.Width := 18;  FButton.Height := 17;  FButton.OnClick := ButtonClick;  FButton.TabStop := False;  FBlackColorMode := False;  LoadEllipsis(FButton.Glyph, False);  FButton.Visible := True;  FFormat := FMT\_DATETIME;  fDateRange := TDateRange.Create;  end; | | |
|  | Modified | procedure TORDateBox.ButtonClick(Sender: TObject);  var  DateDialog: TORDateTimeDlg;  ParsedDate: TFMDateTime;  begin  DateDialog := TORDateTimeDlg.Create(Application);  **DateDialog.DateRange := DateRange;**  **DateDialog.DateTime := DateSelected;**  if Length(Text) > 0 then  begin  ParsedDate := ServerParseFMDate(Text);  if ParsedDate > -1 then  FFMDateTime := ParsedDate  else  FFMDateTime := 0;  end;  DateDialog.DateOnly := FDateOnly;  DateDialog.FMDateTime := FFMDateTime;  DateDialog.RequireTime := FRequireTime;  if DateDialog.Execute then  begin  FFMDateTime := DateDialog.FMDateTime;  UpdateText;  FTimeIsNow := DateDialog.RelativeTime = 'NOW';  **fDateSelected := DateDialog.DateTime;**  end;  DateDialog.Free;  if assigned(OnDateDialogClosed) then  OnDateDialogClosed(self);  if Visible and Enabled then // Some events may hide the component  SetFocus;  end; | | |
|  | Modified | procedure TORDateBox.Validate(var ErrMsg: string);  begin  ErrMsg := '';  if Length(Text) > 0 then  begin  {  !!!!!! THIS HAS BEEN REMOVED AS IT CAUSED PROBLEMS WITH REMINDER DIALOGS - PII !!!!!!  //We need to make sure that there is a date entered before parse  if FRequireTime and ((Pos('@', Text) = 0) or (Length(Piece(Text, '@', 1)) = 0)) then  ErrMsg := 'Date Required';  }  FFMDateTime := ServerParseFMDate(Text);  if FFMDateTime <= 0 then  ErrMsg := 'Invalid Date/Time';  if FRequireTime and (Frac(FFMDateTime) = 0) then  ErrMsg := 'Time Required';  if FDateOnly and (Frac(FFMDateTime) > 0) then  ErrMsg := 'Time not Required';  **if (fDateRange.MinDate > 0) and (FFMDateTime < fDateRange.MinDate) then**  **ErrMsg := 'Date/Time cannot be before ' + FormatDateTime('mm/dd/yyyy',**  **fDateRange.MinDate);**  **if (fDateRange.MaxDate > 0) and (FFMDateTime > fDateRange.MaxDate) then**  **ErrMsg := 'Date/Time cannot be after ' + FormatDateTime('mm/dd/yyyy',**  **fDateRange.MaxDate);**  end;  end; | | |
| **Comments** | New code is shown in **bold** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component** | TORDateCombo | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Description** | Class modified to allow assigning different color to the class components. | | | |
| **Fields** | New | **fColor: TColor** | | |
| **Properties** | New | **property Color: TColor read fColor write SetColor;** | | |
| **Methods** | New | **Procedure TORDateCombo.SetColor(const Value: TColor);**  **begin**  **fColor := Value;**  **FMonthCombo.Color := fColor;**  **FDayCombo.Color := fColor;**  **FYearEdit.Color := fColor;**  **end;** | | |
| **Comments** | New code is shown in **bold** | | | |

###### Identify Required Fields in Text Integration Utility (TIU) Note Templates (20100706)

Summaryof changes

1. The association “GUI element - Template field” is kept in fRequiredControls (uRequiredFieldsHighlighter.pas)
2. A set of routines/objects for handling highlighting and navigation between required fields has been created (files mRequiredFieldsNavigator.pas, uRequiredFieldHighlighter.pas)
3. TTemplateDialog class has been modified to implement highlighting
4. A new frame has been created to localize controls used for navigation between Required fields without values (file mRequiredFieldsNavigator.pas)
5. Additional dialog has been created for selection of the highlighting options (color, location of the navigation frame – files fOptionsTIUTemplates.pas, fOptionsTIUTemplates.dfm)
6. Class TfrmOptions has been modified to let user select highlighting preferences.
7. New RPC was created for saving user preferences in VistA

Files modified

|  |  |
| --- | --- |
| fOptions.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Added button to the “Notes” tab to let user open dialog to specify preferences of highlighting of the required fields in TIU Template. |
| fOptionsTIUTemplates.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  New Dialog implementing selector of the preferable color of highlighted field of the TIU template, location of field navigation buttons |
| uTemplateFields.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Class TTemplateDialogEntry changed to implement highlighting of the required fields without values. |
| fTemplateDialog.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Dialog implementation changed to include support of highlighting and navigation between required fields without value |
| mRequiredFieldsNavigator.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Implementation of the frame containing navigation buttons |
| uRequiredFieldsHighlighter.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Implementation of the highlighting functionality |
| rOptions.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Implementation of the routine loading/saving required fields preferences from/to server |

File fOptions.pas changes

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **TfrmOptions** | fOptions.pas  Modified to implement invocation of the Highlighting preferences dialog TfrmOptionsTIUTemplates | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Controls added** | PnlRequiredFields: TPanel;  Bevel1: TBevel;  StaticText1: TStaticText;  Memo1: TMemo;  btnNotesTitles: TButton;  imgNotes: TImage;  imgRequiredFields: TImage; | | | |
| **Method(s) added** | procedure btnRequiredFieldsClick(Sender: TObject); | | | |
| **Unit(s) references added** | fOptionsTIUTemplates, mRequiredFields; | | | |
| **Class TfrmOptions Modified Logic (New code added)** | | | | |
| procedure TfrmOptions.btnRequiredFieldsClick(Sender: TObject);  begin  UpdateRequiredFieldsPreferences;  end; | | | | |
| **Class TfrmOptions** Current Form | | | | |
| Current layout of the "Notes" tab of the "Options" dialog has no components for control of theTUI required fields settings. | | | | |
| **Class TfrmOptions** Modified Form | | | | |
| Uddated Options dialog includes "Required Fields..." button | | | | |

File fOptionsTIUTemplates changes

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **TfrmTIUTemplates** | File fOptionsTIUTemplates.pas  Implements Highlighting preferences dialog | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form layout** | The "TIU Template" dialog allows user to control settings of the TIU required fields (backgroud color of unpopulated fields and position of the navigation panel) | | | |
| **Class TfrmTIUTemplates Modified Logic (New code added)** | | | | |
| See files fOptionsTIUTemplates.pas; fOptionsTIUTemplates.dfm | | | | |

Components on Form

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| btnOK | TButton | A button, captioned OK, to accept selections on values. |
| btnDefaults | TButton | Resets preferences to the ir default values |
| btnCancel | TButton | Cancels selection, restores the previous values of preferences |
| ckbHighlight | TCheckBox | Enabled/Disables the highlighting |
| rgNavigationPos | TRadioGroup | Position of the navigation bar selector |
| gbHighlightColor | TGroupBox | Group box containing the color selector |
| cbHighlightColor | TColorBox | Color selector |

Events

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| btnDefaultsClick | OnClick | Restores default values of the preferences |
| cbHighlightColorChange | OnChange | Selects color from the drop-down list |
| rgNavigationPosClick | OnClick | Selects position of the navigation frame |
| ckbHighlightClick | OnClick | Enables/Disables the highlighting |

Class Methods

| **Method Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| setDefaults | procedure | Sets default values ov the preferences |
| setInfo | procedure | Info property setter |
| getInfo | function | Info property getter |
| AdjustToFontSize | procedure | Adjusts the form size based on the font size |

Special References

No changes

| **Special Reference Name** | **Type** | **Description** |
| --- | --- | --- |
|  |  |  |

Class Events

No changes

| **Name** | **Type** | **Description** |
| --- | --- | --- |
|  |  |  |

Class Properties

| **Class Properties Name** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| Info | String | Public | String representation of the selected properties |

Unit Variables

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ReqHighlightColor | TColor | Selected highlight color |
| ReqHighlightDisabledColor | TColor | Highlight color for disabled control (default: clSilver) |
| ReqHighlight | Boolean | Highlighting status |
| ReqHighlightAlign | Integer | Location of the navigation buttons |

Unit Methods

| **Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| UpdateRequiredFieldsPreferences | procedure | Opens dialog and saves preferences if changed |
| RestoreHighlightingOptions | procedure | Loads preference string from server |
| SaveHighlightingOptions | procedure | Saves preference string on server |

File uTemplateFields.pas changes

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **TTemplateDialogEntry** | File uTemplateFields.pas  Modifications to implement Highlighting functionality | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Methods Added** | GetControl | | | |
| **Class TTemplateDialogEntry Modified Logic (New code added)** | | | | |
| // NSR20100706 AA 20150706 ----------------------------------------------- begin  function TTemplateDialogEntry.GetControl(CtrlID: integer): TControl;  Var  I: Integer;  Ctrl: TControl;  begin  Result := nil;  for i := 0 to FControls.Count-1 do  begin  Ctrl := TControl(FControls.Objects[i]);  if(assigned(Ctrl)) and (Ctrl.Tag = CtrlID) then  Result := Ctrl;  end;  end;  // NSR20100706 AA 20150706 ------------------------------------------------- end | | | | |

|  |
| --- |
| **Class TTemplateDialogEntry Modified Logic** |
| 1. **Procedure TTemplateField.CreateDialogControls Original code** |
| …  if FSepLines and (FFldType in SepLinesTypes) then  cb.StringData := NewLine;  cb.OnClick := Entry.DoChange;  UpdateColorsFor508Compliance(cb);  inc(Index);  Entry.FControls.InsertObject(Index, '', cb);  if (i=0) or FSepLines then  UpdateIndents(cb);  …  …  if assigned(ctrl) then  begin  inc(Index);  Entry.FControls.InsertObject(Index, '', ctrl);  UpdateIndents(ctrl);  end;  … |
| 1. **Procedure TTemplateField.CreateDialogControls Modified code** |
| …  if FSepLines and (FFldType in SepLinesTypes) then  cb.StringData := NewLine;  cb.OnClick := Entry.DoChange;  UpdateColorsFor508Compliance(cb);  inc(Index);  Entry.FControls.InsertObject(Index, '', cb);  **AddFieldControl(self,cb,IntToStr(ctrlID)); // NSR20100706 AA 2015/10/09. Adding Fld.FID to track Fld by control**  if (i=0) or FSepLines then  UpdateIndents(cb);  …  …  if assigned(ctrl) then  begin  inc(Index);  Entry.FControls.InsertObject(Index, '', ctrl);  **addFieldControl(self,TWinControl(ctrl),IntToStr(ctrlID)); // NSR20100706 AA 2015/10/09 adding Fld.FID to track Fld by control**  UpdateIndents(ctrl);  end;  … |

Components on Form

N/A

Events

No changes

Special References

No changes

Class Events

No changes

Unit Methods

| **Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| getUTmplFlds | function | Access to the uTmplFlds string list |
| function getUTmplFlds:TList; // - access to template fields NSR20100706 AA 2015/10/07  begin  Result := uTmplFlds;  end; | | |

Class Properties

No changes

Uses Clause

|  |  |
| --- | --- |
| **Unit(s) references added to interface section** | ORNet |
| **Unit(s) references added to implementation section** | fTemplateDialog, uRequiredFieldsHighlighter; |

Changes to fTemplateDialog.pas

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| fTemplateDialog.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Dialog implementation changed to include support of highlighting and navigation between required fields without value | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Controls added** | frRequiredFields: TTRequiredFieldsFrame;  RFH: TRequiredFieldsHighlighter | | | |
|  |  | | | |
| **Method(s) added** | 1. procedure btnRequiredFieldsClick(Sender: TObject); 2. procedure FieldValueChanged(Sender: TObject); procedure CMFocusChanged(var Message: TCMFocusChanged); Message CM\_FocusChanged; procedure UpdateControlHighlighting(ctrl:TWinControl); 3. procedure setOKStatus; 4. procedure SetControlDebugInfo(aControl:TWinControl); 5. procedure SetDebugInfo; 6. procedure setReqHighlightAlign(aPos:Integer); procedure setReqHighlightColor; 7. procedure AdjustFormToFontSize(aSize:Integer); | | | |
| **Unit(s) references added** | fOptionsTIUTemplates, mRequiredFieldsNavogator; | | | |

Current Form

|  |
| --- |
| **Class TfrmTemplateDialog Current Form** |
| Current layout of the "TIU Template" dialog does limits indication of required fields with "*" symbol. |

Modified Form

| **Class TfrmTemplateDialog Current Form** |
| --- |
| Updated dialog indicates unpopulated required fields with different background. A naviagtion panel offers buttons for qick access to unpopulated fields.  **TIU Temmplate dialog. Updated layout**  **Updated dialog indicates unpopulated required fields with different background. A naviagtion panel offers buttons for qick access to unpopulated fields.** |

Components on Form

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| frRequiredFields | TRequiredFieldsFrame | Frame encapsulated navigation controls |
| pnlBottomGrid | TPanel | Container of buttons for general function |
| grdpnlBottom | TGridPanel | Container of dialog controls |
| pnlBottomGrid | TPanel | Bottom margin of the navigation panel |
| pnlBottomLeft | TPanel | Left margin of the navigation panel |
| pnlBottomRight | TPanel | Right margin of the navigation panel |
| pnlBtnTop | TPanel | Top margin of the navigation panel |
| PopupMenu1 | TPopupMenu | Popup menu of the form |
| T1 | TMenuItem | Popup menu option |
| S1 | TMenuItem | Popup menu option |
| S4 | TMenuItem | Popup menu option |
| splFields | TSplitter | Splitter |
| pnlDebug | TPanel | Container for debug controls |
| reText | TRichEdit | Container of the debug information |

Events

No changes

| **Name** | **Type** | **Description** |
| --- | --- | --- |
|  |  |  |

Methods

No changes

| **Method Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
|  |  |  |

Special References

| **Special Reference Name** | **Type** | **Description** |
| --- | --- | --- |
| RFH | TRequiredFieldsHighlighter | Object handling list of known template controls |

Class Events

No changes

| **Name** | **Type** | **Description** |
| --- | --- | --- |
|  |  |  |

Class Methods

| **Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| FieldValueChanged | procedure | Handler of OnChange event for a control |
| CMFocusChanged | procedure | CM\_FocusChanged message handler |
| updateControlHighlighting | procedure | Highlighting focus control |
| setOKStatus | procedure | Updating enabled status of the OK button based on number of unpopulated required fields |
| SetControlDebugInfo | Procedure | Updates debug info of focused control |
| setDebugInfo | procedure | Updates debug info |
| setReqHighlightAlign | procedure | Setting position of the navigation frame |
| setReqHighlightColor | procedure | Updating color of highlighting |
| AdjustFormToFontSize | procedure | Adjusting size of controls based on the selected font |

Class Properties

No changes

| **Class Properties Name** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
|  |  |  |  |

Uses Clause

|  |  |
| --- | --- |
| **Unit reference(s) added to interface section** | mRequiredFieldsNavigator, Vcl.ComCtrls, Vcl.Menus |
| **Unit reference(s) added to implementation section** | fOptionsTIUTemplates, ORNet |

New Code added

|  |
| --- |
| **Class Class TfrmTemplateDialog**  **Modified Logic (New code added)** |

// NSR20100706 AA 2015/09/29 --------------------------------------------- begin

procedure TfrmTemplateDialog.FieldValueChanged(Sender: TObject);

begin

UpdateControlHighlighting(ActiveControl);

end;

procedure TfrmTemplateDialog.S1Click(Sender: TObject);

begin

inherited;

S1.Checked := not S1.Checked;

ReqHighlight := S1.Checked;

RFH.HighlightControls(ReqHighlight);

end;

procedure TfrmTemplateDialog.S4Click(Sender: TObject);

begin

inherited;

UpdateRequiredFieldsPreferences;

frRequiredFields.setAlign(ReqHighlightAlign);

end;

const

ss = ' -----------------------------------------------------------------------';

procedure TfrmTemplateDialog.CMFocusChanged(var Message: TCMFocusChanged);

begin

{$IFDEF DEBUG}

Caption :=

TWinControl(Message.Sender).Name + ' ' +

TWinControl(Message.Sender).ClassName;

{$ENDIF}

UpdateControlHighlighting(TWinControl(Message.Sender));

if ActiveControl is TORComboEdit then

// UpdateNavigation(ActiveControl.Parent) // to address TORComboEdit

frRequiredFields.FocusedControl := ActiveControl.Parent // to address TORComboEdit

else

// UpdateNavigation(ActiveControl) // to address TORComboEdit

frRequiredFields.FocusedControl := ActiveControl // to address TORComboEdit

end;

procedure TfrmTemplateDialog.UpdateControlHighlighting(ctrl:TWinControl);

begin

if not RFH.IsTemplateControl(ctrl,sbMain) then

Exit;

// HighlightControlInstance(ctrl,ReqHighlight); // doesn't update empty TORComboEdit :(

frRequiredFields.CurrentControl := ctrl;

frRequiredFields.RequiredTotal := RFH.getNumberOfMissingFields(sbMain);

RFH.HighlightControls(ReqHighlight);

setOKStatus;

{$IFDEF DEBUG}

setControlDebugInfo(ctrl);

{$ENDIF}

end;

procedure addSL(aTitle:String;aSL,aTarget:TStrings);

begin

if not Assigned(aTarget) then

exit;

if not Assigned(aSL) then

begin

aTarget.Add(aTitle+ ': NOT ASSIGNED ');

aTarget.Add('');

end

else

begin

aTarget.Add(copy(aTitle+ ' ' + ss,1,74)+ ' begin');

if aSL.Text <> '' then

begin

aTarget.Text := aTarget.Text +aSL.Text;

if trim(aTarget[aTarget.Count-1])='' then

aTarget.Delete(aTarget.Count-1);

end;

aTarget.Add(copy(aTitle+ ' ' + ss,1,76)+ ' end');

aTarget.Add('');

end;

end;

procedure TfrmTemplateDialog.setReqHighlightColor;

begin

try

RFH.HighlightControls(ReqHighlight);

except

on E: Exception do

ShowMessage(E.Message);

end;

end;

procedure TfrmTemplateDialog.setOKStatus;

begin

btnOKGrid.Enabled :=

RFH.getNumberOfMissingFields(sbMain)<1;

end;

procedure TfrmTemplateDialog.setReqHighlightAlign(aPos:Integer);

begin

try

frRequiredFields.setAlign(aPos);

except

on E: Exception do

ShowMessage(E.Message);

end;

end;

procedure TfrmTemplateDialog.AdjustFormToFontSize(aSize:Integer);

var

W: Integer;

procedure adjustWidthToFont(aSize:Integer; var X:Integer);

begin

case aSize of

8,9:

X := 800;

10,11:

X := 800;

12,13:

X := 840;

14,15,16,17:

X := 1020;

18:

X := 1200;

else

X := 1280;

end;

end;

begin

adjustWidthToFont(aSize,w);

Width := w;

Constraints.MinWidth := w;

pnlBottomGrid.Height := frRequiredFields.szButtonY +

pnlBtnTop.Height + pnlBtnBottom.Height;

grdpnlBottom.ColumnCollection[0].Value := frRequiredFields.szButtonX;

grdpnlBottom.ColumnCollection[1].Value := frRequiredFields.szButtonX;

grdpnlBottom.ColumnCollection[3].Value := frRequiredFields.szButtonX;

grdpnlBottom.ColumnCollection[5].Value := frRequiredFields.szButtonX;

grdpnlBottom.ColumnCollection[6].Value := frRequiredFields.szButtonX;

pnlBottomRight.Width := frRequiredFields.pnlRMargin.Width;

pnlBottomLeft.Width := frRequiredFields.pnlLMargin.Width;

end;

procedure TfrmTemplateDialog.T1Click(Sender: TObject);

begin

inherited;

T1.Checked := not T1.Checked;

{$IFDEF DEBUG}

SetDebugInfo;

{$ENDIF}

end;

{$IFDEF DEBUG}

procedure TfrmTemplateDialog.SetControlDebugInfo(aControl:TWinControl); //NSR20100706 AA 2015/09/29

var

sValue: String;

fld: TTemplateField;

begin

fld := RFH.getTemplateFieldByControl(aControl);

reText.Clear;

reText.Lines.Add(Format('Field: %d',[Integer(fld)]));

if assigned(fld) then

begin

reText.Lines.Add(' ID: '+fld.ID);

reText.Lines.Add(' Name: '+fld.FldName);

reText.Lines.Add(' Value: '+sValue);

if Fld.Required then

reText.Lines.Add(' Required Field')

else

reText.Lines.Add(' Optional Field');

end

else

reText.Lines.Add(' NO DATA ASSIGNED');

reText.Lines.Add(Format('Control: %d',[Integer(aControl)]));

reText.Lines.Add(' Class: '+aControl.ClassName);

reText.Lines.Add(' Tag: '+IntToStr(aControl.Tag));

reText.Lines.Add(' Info: '+RFH.getControlInfo(aControl));

end;

procedure TfrmTemplateDialog.setDebugInfo;

var

SL: TStringList;

lst:TList;

i: Integer;

procedure setDebugLayout;

var

fDebug: Boolean;

begin

fDebug := T1.Checked;

pnlDebug.Visible := fDebug;

if not fDebug then

begin

splFields.Align := alTop;

splFields.Visible := False;

end

else

begin

splFields.Visible := True;

splFields.Align := alBottom;

end;

end;

procedure AddControlsInfo(aSL: TStringList;aCtrl: TWinControl;anOffset:String =' ');

var

i: integer;

begin

if not assigned(aCtrl) then

exit;

SL.Add(anOffset+ Format('%10.10d %10.10d %20.20s %s',[Integer(aCtrl),aCtrl.Tag,aCtrl.Name,aCtrl.ClassName]));

for I := 0 to aCtrl.ControlCount -1 do

if aCtrl.Controls[i] is TWinControl then

AddControlsInfo(aSL,TWinControl(aCtrl.Controls[i]),anOffset);

end;

function toSL:TSTringList;

begin

Result := TStringList.Create;

Result.Add(Format('Class: %s Item: %d',[ClassName,Integer(self)]));

Result.Add('Name: '+Name);

Result.Add('Index: '+Index);

Result.Add('Count: '+IntToStr(Count));

Result.Add('Font Size: '+IntToStr(Font.Size));

if fFirstBuild then

Result.Add('FirstBuild: '+'True')

else

Result.Add('FirstBuild: '+'False');

AddSL('BuildIdx:',BuildIdx,Result);

AddSL('NoTextID:',NoTextID,Result);

AddSL('Entries:',Entries,Result);

end;

begin

reText.Clear;

SL := toSL;

reText.Text := SL.Text;

reText.Text := reText.Text + CRLF +

'Required Fields Navigation:'+ CRLF+

frRequiredFields.toStr + CRLF;

AddSL('ControlList:',RFH.getRequiredControls,reText.Lines);

SL.Clear;

AddControlsInfo(SL,sbMain);

AddSL('sbMain.Controls:',SL,reText.Lines);

SL.Clear;

lst := getuTmplFlds;

for i := 0 to lst.Count -1 do

SL.Add(Format('%d %20.20s',[i,TTemplateField(lst[i]).ID]));

AddSL('uTmplFlds:',SL,reText.Lines);

SL.Free;

SetDebugLayout;

end;

{$ENDIF}

// NSR20100706 AA 2015/09/29 ----------------------------------------------- end

|  |
| --- |
| Class TfrmTemplateDialogEntry Current Logic (procedure TfrmTemplateDialog.FieldChanged(Sender: TObject)) |
| procedure TfrmTemplateDialog.FieldChanged(Sender: TObject);  begin  with TTemplateDialogEntry(Sender) do  TPanel(Obj).hint := GetText;  end; |
| Class TfrmTemplateDialog Modified Logic (procedure TfrmTemplateDialog.FieldChanged(Sender: TObject)) |
| procedure TfrmTemplateDialog.FieldChanged(Sender: TObject);  begin  with TTemplateDialogEntry(Sender) do  TPanel(Obj).hint := GetText;  **fieldValueChanged(Sender); // notifying dialog the value have changed NSR20100706 AA 2015/09/29**  end; |
| Class TfrmTemplateDialogEntry Current Logic (procedure TfrmTemplateDialog.FormeClosed) |
| procedure TfrmTemplateDialog.FormClose(Sender: TObject;  var Action: TCloseAction);  begin  Application.OnShowHint := FOldHintEvent;  SaveUserBounds(Self);  end; |
| Class TfrmTemplateDialog Modified Logic (procedure TfrmTemplateDialog.FormClosed) |
| procedure TfrmTemplateDialog.FormClose(Sender: TObject;  var Action: TCloseAction);  begin  Application.OnShowHint := FOldHintEvent;  SaveUserBounds(Self);  **saveHighlightOptions; // NSR20100706 AA 2015/09/29**  end; |

|  |
| --- |
| Class TfrmTemplateDialog Current Logic (procedure TfrmTemplateDialog.FormCloseQuery(Sender: TObject; var CanClose: Boolean);) |
| procedure TfrmTemplateDialog.FormCloseQuery(Sender: TObject; var CanClose: Boolean);  var  Txt, tmp: string;  i, p1, p2: Integer;  Save: Boolean;  begin  CanClose := TRUE;  if FCheck4Required then  begin  FCheck4Required := FALSE;  Txt := SL.Text;  for i := 0 to sbMain.ControlCount - 1 do  begin  Save := FALSE;  if (sbMain.Controls[i] is TORCheckBox) and  (TORCheckBox(sbMain.Controls[i]).Checked) then  Save := TRUE  else  if (OneOnly and (sbMain.Controls[i] is TPanel)) then  Save := TRUE;  if (Save) then  begin  tmp := Piece(Index,U,sbMain.Controls[i].Tag);  p1 := StrToInt(Piece(tmp, '~', 1));  p2 := StrToInt(Piece(tmp, '~', 2));  if AreTemplateFieldsRequired(copy(Txt, p1, p2)) then  CanClose := FALSE;  end;  end;  if not CanClose then  begin  ShowMsg(MissingFieldsTxt);  break;  end;  end;  end; |

|  |
| --- |
| Class TfrmTemplateDialog Modified Logic (procedure TfrmTemplateDialog.FormCloseQuery(Sender: TObject; var CanClose: Boolean);) |

procedure TfrmTemplateDialog.FormCloseQuery(Sender: TObject; var CanClose: Boolean);

var

Txt, tmp: string;

i, p1, p2: Integer;

Save: Boolean;

begin

CanClose := TRUE;

if FCheck4Required then

begin

**// NSR2010706 enables OK button only if all the required fields are populated.**

**// Since FCheck4Required is set TRUE by OK button this code won't be called.**

**// Leaving "as is" in case it is needed for reminder dialogs processing**

FCheck4Required := FALSE;

Txt := SL.Text;

for i := 0 to sbMain.ControlCount - 1 do

begin

Save := FALSE;

if (sbMain.Controls[i] is TORCheckBox) and

(TORCheckBox(sbMain.Controls[i]).Checked) then

Save := TRUE

else

if (OneOnly and (sbMain.Controls[i] is TPanel)) then

Save := TRUE;

if (Save) then

begin

tmp := Piece(Index,U,sbMain.Controls[i].Tag);

p1 := StrToInt(Piece(tmp, '~', 1));

p2 := StrToInt(Piece(tmp, '~', 2));

if AreTemplateFieldsRequired(copy(Txt, p1, p2)) then

CanClose := FALSE;

end;

end;

if not CanClose then

begin **// NSR20100706 AA -------------------------------------------- begin**

**ReqHighlight := True;**

**frRequiredFields.Visible := True;**

**RFH.HighlightControls(ReqHighlight);**

**{$IFDEF DEBUG}**

**setDebugInfo;**

**{$ENDIF}**

**txt := RFH.getRequiredFieldNames(sbMain,char(VK\_TAB));**

**i := pos(CRLF,txt);**

**if i < (Length(txt)-Length(CRLF)+1) then**

**txt := 'The following Required fields are not populated:'+ CRLF+CRLF +txt**

**else**

**txt := 'The following Required field is not populated:'+ CRLF+CRLF +txt;**

**frRequiredFields.FocusControl; // scroll to the first unpopulated field**

**ShowMsg(txt);**

**end; // NSR20100706 AA ----------------------------------------------** end

end;

end;

|  |
| --- |
| Class TfrmTemplateDialog Current Logic (procedure TfrmTemplateDialog.FormClose(Sender: TObject;) |
| procedure TfrmTemplateDialog.FormClose(Sender: TObject;  var Action: TCloseAction);  begin  Application.OnShowHint := FOldHintEvent;  SaveUserBounds(Self);  end; |
| Class TfrmTemplateDialog Modified Logic (procedure TfrmTemplateDialog.FormClose(Sender: TObject;) |
| procedure TfrmTemplateDialog.FormClose(Sender: TObject;  var Action: TCloseAction);  begin  Application.OnShowHint := FOldHintEvent;  SaveUserBounds(Self);  **saveUserOptions; // NSR20100706 AA 2015/09/29**  end; |

Frame mTemplateFieldButton changes

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **TfraTemplateFieldButton** | File mTemplateFieldButton.pas  Implements Highlighting of template field of type “Button”  Modified to include TShape component providing color to the button.  Code added to update button size to match caption width | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Old code** | procedure TfraTemplateFieldButton.SetButtonText(const Value: string);  begin  lblText.Caption := Value;  end; | | | |
| **New code** | procedure TfraTemplateFieldButton.SetButtonText(const Value: string);  var  i: Integer;  const  iGap = 4;  begin  lblText.Caption := Value;  i := lblText.Canvas.textWidth(Value);  if width < i + iGap then  width := i + iGap ;  end; | | | |

File mRequiredFieldsNavigatorpas changes

| **Class** | | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- | --- |
| **TRequiredFieldsFrame** | | File mRequiredFieldsNavigator.pas  Implements Highlighting navigation buttons frame | | | |
| **Enhancement Category** | | New | Modify | Delete | No Change |
| **Frame layout** | The navigation panel contains buttons for quick access to unpopulated required fields of the template.  Unpopulated Fields navigation panel.  The navigation panel contains buttons for quick access to unpopulated required fields of the template. | | | | |
| **Class TRequiredFieldsFrame Modified Logic (New code added)** | | See file mRequiredFieldsNavigator.pas; | | | |

Components on Form

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| gpButtons | TGridPanel | Navigation buttons container |
| btnPrev | TButton | Navigation button |
| btnNext | TButton | Navigation button |
| btnFirst | TButton | Navigation button |
| btnLast | TButton | Navigation button |
| stxtTotalRequired | TStaticText | Indicator of the total number of unpopulated required fields |
| PopupMenu1 | TPopupMenu | Context menu |
| T1 | TMenuItem | Context menu item |
| B1 | TMenuItem | Context menu item |
| L1 | TMenuItem | Context menu item |
| R1: | TMenuItem | Context menu item |
| pnlLmargin | TPanel | Left margin panel |
| pnRLmargin | TPanel | Right margin panel |

Special References

No references

| **Special Reference Name** | **Type** | **Description** |
| --- | --- | --- |
| RFH | TRequiredFieldsHighlighter | Manager of the association “Template Fields – form control” |

Class Events

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| btnFirstClick | onClick | Processing of the navigation button clicks |
| T1Click | OnClick | Processing menu button clicks |

Class Methods

| **Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| setRequiredTotal(aValue: Integer) | procedure | Processes changes in conter of the required fields vithout values |
| setCurrentControl(aControl: TWinControl); | procedure |  |
| setFocusedControl(aControl: TWinControl) | procedure |  |
| getCtrlList: TStringList | function |  |
|  |  |  |
| toStr:String | function | Debug info |
| getNextRequiredControlInd( aCurrent: Integer; aDirection:Integer):Integer | function | Index of the next control |
| setButtonStatus | procedure | Set status of the OK button |
| setAlign(anAl:TAlign); overload | procedure | Sets alignment of the frame |
| setAlign(anAl:Integer); overload | procedure | Sets alignment of the frame |
| focusControl(anID:Integer=-1); | procedure | Sets focus on control by control ID |
| adjustButtonSize(aSize:Integer); | procedure | Adjusts size of the buttons |
| setRowColumn(aRow,aColumn: Integer; aControlItem: TControlItem); | procedure | Assigns a control to position in TGridPanel |
| setV(aHeight,aWidth:Integer); | procedure | Sets TGridPanel properties for framealign to the Left or Right |
| setH(aHeight,aWidth:Integer); | procedure | Sets TGridPanel properties for framealign to the Top or Bottom |
|  |  |  |

Class Properties

| **Class Properties Name** | **Type** | **Visibility** | **Description** |
| --- | --- | --- | --- |
| CtrlList | TStringList | public |  |
| CurrentControl: TWinControl read fCurrentControl write setCurrentControl | TWinControl | public | Pointer to the control having focus |
| CurrentControlInd: Integer read fCurrentControlInd write fCurrentControlInd; | Integer | public | Index of the control with focus in the list of template controls |
| property RequiredTotal: Integer read fRequiredTotal write setRequiredTotal | Integer | public | Number of the unpopulated required fields |
| FocusedControl | TWinControl | public |  |
|  |  |  |  |

New Code added

|  |
| --- |
| **Class Class TRequiredFieldsFrame**  **Modified Logic (New code)** |

unit mRequiredFieldsNavigator;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants,

System.Classes,

Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls,

Vcl.ComCtrls, Vcl.ExtCtrls, uTemplateFields, TypInfo, ORFn, ORNet, StrUtils,

fxBroker, ORCtrls, ORDtTm, Vcl.Menus, uRequiredFieldsHighlighter;

type

TRequiredFieldsFrame = class(TFrame)

gpButtons: TGridPanel;

btnPrev: TButton;

btnNext: TButton;

btnFirst: TButton;

btnLast: TButton;

stxtTotalRequired: TStaticText;

PopupMenu1: TPopupMenu;

T1: TMenuItem;

B1: TMenuItem;

L1: TMenuItem;

R1: TMenuItem;

pnlLmargin: TPanel;

pnlRMargin: TPanel;

procedure btnFirstClick(Sender: TObject);

procedure T1Click(Sender: TObject);

private

{ Private declarations }

fCurrentControl: TWinControl;

fFocusedControl: TWinControl;

fCurrentControlInd: Integer;

fRequiredTotal: Integer;

procedure setRequiredTotal(aValue: Integer);

procedure setCurrentControl(aControl: TWinControl);

procedure setFocusedControl(aControl: TWinControl);

function getCtrlList: TStringList;

public

{ Public declarations }

szButtonX: Integer;

szButtonY: Integer;

szMarginH: Integer;

szMarginV: Integer;

RFH: TRequiredFieldsHighlighter;

property CtrlList: TStringList read getCtrlList;

property CurrentControl: TWinControl read fCurrentControl

write setCurrentControl;

property CurrentControlInd: Integer read fCurrentControlInd

write fCurrentControlInd;

property RequiredTotal: Integer read fRequiredTotal write setRequiredTotal;

property FocusedControl: TWinControl read fFocusedControl

write setFocusedControl;

function toStr: String;

function getNextRequiredControlInd(aCurrent: Integer;

aDirection: Integer): Integer;

procedure setButtonStatus;

procedure setAlign(anAl: TAlign); overload;

procedure setAlign(anAl: Integer); overload;

procedure focusControl(anID: Integer = -1);

procedure adjustButtonSize(aSize: Integer);

procedure setRowColumn(aRow, aColumn: Integer; aControlItem: TControlItem);

procedure setV(aHeight, aWidth: Integer);

procedure setH(aHeight, aWidth: Integer);

end;

procedure adjustButtonSizeToFont(aSize: Integer; var X, Y, HGap, VGap: Integer);

implementation

{$R \*.dfm}

uses

VAUtils, uDlgComponents, mTemplateFieldButton, rMisc, fOptionsTIUTemplates;

procedure adjustButtonSizeToFont(aSize: Integer; var X, Y, HGap, VGap: Integer);

begin

HGap := 8;

case aSize of

8, 9:

begin

X := 75;

Y := 24;

VGap := 8;

end;

10, 11:

begin

X := 85;

Y := 26;

VGap := 10;

end;

12, 13:

begin

X := 95;

Y := 32;

VGap := 12;

end;

14, 15, 16, 17:

begin

X := 120;

Y := 38;

VGap := 16;

end;

18:

begin

X := 150;

Y := 42;

VGap := 20;

end;

else

begin

X := 105;

Y := 44;

end;

end;

end;

/// /////////////////////////////////////////////////////////////////////////////

function TRequiredFieldsFrame.getCtrlList: TStringList;

begin

Assert(RFH <> nil, 'RFH is not assigned');

Result := RFH.getRequiredControls;

end;

procedure TRequiredFieldsFrame.setButtonStatus;

var

i: Integer;

function getFirstRequiredControlIndex: Integer;

var

ind: Integer;

ctrl: TWinControl;

fld: TTemplateField;

begin

Result := -1;

for ind := 0 to CtrlList.Count - 1 do

begin

ctrl := TWinControl(CtrlList.Objects[ind]);

if ctrl is TCPRSDialogParentCheckBox then

continue;

if pos(U, CtrlList[ind]) = 0 then

continue;

fld := RFH.getTemplateFieldByControl(ctrl);

if assigned(fld) and fld.Required then

begin

Result := ind;

break;

end;

end;

end;

function getLastRequiredControlIndex: Integer;

var

ind: Integer;

ctrl: TWinControl;

fld: TTemplateField;

begin

Result := -1;

for ind := CtrlList.Count - 1 downto 0 do

begin

ctrl := TWinControl(CtrlList.Objects[ind]);

if ctrl is TCPRSDialogParentCheckBox then

continue;

if pos(U, CtrlList[ind]) = 0 then

continue;

fld := RFH.getTemplateFieldByControl(ctrl);

if assigned(fld) and fld.Required then

begin

Result := ind;

break;

end;

end;

end;

begin

Assert(RFH <> nil, 'RFH is not assigned');

case fRequiredTotal of

0:

begin

btnFirst.Enabled := False;

btnLast.Enabled := False;

btnPrev.Enabled := False;

btnNext.Enabled := False;

end;

1:

begin

btnFirst.Enabled := True;

btnLast.Enabled := False;

btnPrev.Enabled := False;

btnNext.Enabled := False;

end;

else

begin

i := getNextRequiredControlInd(CurrentControlInd, -1);

btnFirst.Enabled := (RequiredTotal > 0) and

(CurrentControlInd <> getFirstRequiredControlIndex);

btnPrev.Enabled := i >= 0;

i := getNextRequiredControlInd(CurrentControlInd, 1);

btnNext.Enabled := (RequiredTotal > 0) and

(i <= RFH.getRequiredControls.Count - 1);

btnLast.Enabled := btnNext.Enabled;

end;

end;

end;

procedure TRequiredFieldsFrame.setRequiredTotal(aValue: Integer);

var

s: String;

begin

fRequiredTotal := aValue;

s := 'All Required Fields have Values';

if aValue <> 0 then

s := 'Total Number of Required Fields without Values: ' +

Format('%d', [aValue]);

stxtTotalRequired.Caption := s;

setButtonStatus;

end;

procedure TRequiredFieldsFrame.T1Click(Sender: TObject);

begin

if Sender = T1 then

setAlign(alTop)

else if Sender = B1 then

setAlign(alBottom)

else if Sender = L1 then

setAlign(alLeft)

else if Sender = R1 then

setAlign(alRight)

end;

function TRequiredFieldsFrame.getNextRequiredControlInd(aCurrent: Integer;

aDirection: Integer): Integer;

var

ctrl: TWinControl;

sFLD, sFLDCurrent: String;

bFound: Boolean;

ind: Integer;

fld: TTemplateField;

begin

Assert(RFH <> nil, 'RFH is not assigned');

Result := aCurrent;

ind := aCurrent;

if not assigned(CtrlList) then

exit;

sFLDCurrent := '';

if (ind >= 0) and (ind < CtrlList.Count) then

begin

ind := ind - aDirection;

repeat

inc(ind, aDirection);

if not(CtrlList.Objects[ind] is TCPRSDialogParentCheckBox) then

break;

until (ind = 0) or (ind = CtrlList.Count - 1);

if ind = CtrlList.Count then

exit;

CurrentControlInd := ind;

sFLDCurrent := RFH.parseControlFLD(CtrlList[CurrentControlInd]);

end;

bFound := False;

repeat

ind := ind + aDirection;

if ind < 0 then

bFound := True

else if ind >= CtrlList.Count then

bFound := True

else

begin

ctrl := TWinControl(CtrlList.Objects[ind]);

if ctrl is TCPRSDialogParentCheckBox then

continue;

if pos(U, CtrlList[ind]) = 0 then

continue;

sFLD := RFH.parseControlFLD(CtrlList[ind]);

if sFLD = sFLDCurrent then

continue;

fld := RFH.getTemplateFieldByControl(ctrl);

if not assigned(fld) or not fld.Required then

continue;

if not TWinControl(CtrlList.Objects[ind]).Enabled then

continue;

if RFH.getRequiredFieldText(TWinControl(CtrlList.Objects[ind])) <> '' then

continue;

bFound := True;

end;

until bFound;

Result := ind;

end;

procedure TRequiredFieldsFrame.btnFirstClick(Sender: TObject);

var

tmpBn: TButton;

iPos, i: Integer;

function getFLDFirstControlInd(aFLD: String): Integer;

var

i: Integer;

s: String;

begin

Result := -1;

s := aFLD;

for i := 0 to CtrlList.Count - 1 do

if RFH.parseControlFLD(CtrlList[i]) = s then

break;

if (i >= 0) and (i < CtrlList.Count) then

Result := i;

end;

begin

inherited;

Assert(RFH <> nil, 'RFH is not assigned');

i := RFH.getNumberOfMissingFields(parent); // assuming parent is sbMain

if i > 0 then

begin

iPos := -1;

tmpBn := (Sender as TButton);

if tmpBn = btnFirst then

iPos := getNextRequiredControlInd(-1, 1)

else if tmpBn = btnLast then

iPos := getNextRequiredControlInd(CtrlList.Count, -1)

else if tmpBn = btnPrev then

iPos := getNextRequiredControlInd(CurrentControlInd, -1)

else if tmpBn = btnNext then

iPos := getNextRequiredControlInd(CurrentControlInd, 1);

if (iPos >= 0) and (iPos < CtrlList.Count) then

CurrentControlInd := iPos;

if (CurrentControlInd < 0) or (CurrentControlInd >= CtrlList.Count) then

else if (CtrlList.Objects[CurrentControlInd] is TWinControl) then

begin

if TWinControl(CtrlList.Objects[CurrentControlInd]).Visible and

TWinControl(CtrlList.Objects[CurrentControlInd]).Enabled then

TWinControl(CtrlList.Objects[CurrentControlInd]).SetFocus;

stxtTotalRequired.Caption :=

IntToStr(Integer(CtrlList.Objects[CurrentControlInd])) + ' ' +

CtrlList.Objects[CurrentControlInd].ClassName;

end;

end;

RequiredTotal := i;

end;

function TRequiredFieldsFrame.toStr: String;

begin

Result := Format('Current Control Index: %d', [CurrentControlInd]);

Result := Result + CRLF + 'Font Size: ' + IntToStr(Font.Size) + CRLF +

'Button Width: ' + IntToStr(szButtonX) + CRLF + 'Button Height: ' +

IntToStr(szButtonY) + CRLF + 'Button Margin: ' + IntToStr(szMarginH) + CRLF

+ 'Width: ' + IntToStr(Width) + CRLF + 'Heidth: ' + IntToStr(Height) + CRLF;

Result := Result + CRLF + RFH.getRequiredFieldNames(parent, ' ');

end;

procedure TRequiredFieldsFrame.focusControl(anID: Integer = -1);

var

ind: Integer;

begin

if CtrlList.Count < 1 then

exit;

if anID = -1 then

ind := CurrentControlInd

else

ind := anID;

if ind < 0 then

ind := 0;

if assigned(CtrlList.Objects[ind]) then

if assigned(TWinControl(CtrlList.Objects[ind]).parent) and

TWinControl(CtrlList.Objects[ind]).Enabled then

TWinControl(CtrlList.Objects[ind]).SetFocus

else

ShowMessage('Error setting focus on control ' + CRLF +

TWinControl(CtrlList.Objects[ind]).Name + CRLF +

TWinControl(CtrlList.Objects[ind]).QualifiedClassName);

end;

procedure TRequiredFieldsFrame.setCurrentControl(aControl: TWinControl);

var

ind: Integer;

begin

if aControl = nil then

exit;

if aControl is TCPRSDialogParentCheckBox then

exit;

if aControl is TORComboEdit then

exit;

ind := getCtrlList.IndexOfObject(aControl);

CurrentControlInd := ind;

end;

procedure TRequiredFieldsFrame.setFocusedControl(aControl: TWinControl);

var

ind: Integer;

fld: TTemplateField;

begin

Assert(RFH <> nil, 'RFH is not assigned');

fFocusedControl := aControl;

fld := RFH.getTemplateFieldByControl(aControl);

if assigned(fld) then

begin

if fld.Required then

if (RFH.getRequiredFieldText(fld, U) = '') then

CurrentControl := aControl

else

begin

ind := RFH.getControlIndex(aControl);

if ind > -1 then

begin

ind := getNextRequiredControlInd(ind, 1);

if (ind > -1) and (ind < RFH.getRequiredControls.Count) then

CurrentControl := RFH.getControlByInd(ind);

end;

end;

setButtonStatus;

end;

end;

procedure TRequiredFieldsFrame.setRowColumn(aRow, aColumn: Integer;

aControlItem: TControlItem);

begin

aControlItem.Column := aColumn;

aControlItem.Row := aRow;

aControlItem.RowSpan := 1;

aControlItem.ColumnSpan := 1;

end;

procedure TRequiredFieldsFrame.setV(aHeight, aWidth: Integer);

var

i: Integer;

ctrl: TControlItem;

ci: TCellItem;

begin

ci := gpButtons.RowCollection[0];

ci.SizeStyle := ssAbsolute;

ci.Value := 4.0;

for i := 1 to 4 do

begin

ci := gpButtons.RowCollection[i];

ci.SizeStyle := ssAbsolute;

ci.Value := aHeight;

end;

ci := gpButtons.RowCollection[5];

ci.SizeStyle := ssPercent;

ci.Value := 100.0;

ci := gpButtons.ColumnCollection[0];

ci.SizeStyle := ssPercent;

ci.Value := 100.0;

for i := 1 to 4 do

begin

ci := gpButtons.ColumnCollection[i];

ci.SizeStyle := ssAbsolute;

ci.Value := 0.0;

end;

for i := 0 to gpButtons.ControlCollection.Count - 1 do

begin

ctrl := gpButtons.ControlCollection[i];

if ctrl.Control = btnFirst then

setRowColumn(1, 0, ctrl)

else if ctrl.Control = btnPrev then

setRowColumn(2, 0, ctrl)

else if ctrl.Control = btnNext then

setRowColumn(3, 0, ctrl)

else if ctrl.Control = btnLast then

setRowColumn(4, 0, ctrl)

else if ctrl.Control = stxtTotalRequired then

setRowColumn(5, 0, ctrl)

end;

end;

procedure TRequiredFieldsFrame.setH(aHeight, aWidth: Integer);

var

i: Integer;

ci: TCellItem;

ctrl: TControlItem;

begin

ci := gpButtons.RowCollection[0];

ci.SizeStyle := ssAbsolute;

ci.Value := aHeight;

for i := 1 to 5 do

begin

ci := gpButtons.RowCollection[i];

ci.SizeStyle := ssAbsolute;

ci.Value := 0.0;

end;

ci := gpButtons.ColumnCollection[0];

ci.SizeStyle := ssPercent;

ci.Value := 100.0;

for i := 1 to 4 do

begin

ci := gpButtons.ColumnCollection[i];

ci.SizeStyle := ssAbsolute;

ci.Value := aWidth;

end;

for i := 0 to gpButtons.ControlCollection.Count - 1 do

begin

ctrl := gpButtons.ControlCollection[i];

if ctrl.Control = stxtTotalRequired then

setRowColumn(0, 0, ctrl)

else if ctrl.Control = btnFirst then

setRowColumn(0, 1, ctrl)

else if ctrl.Control = btnPrev then

setRowColumn(0, 2, ctrl)

else if ctrl.Control = btnNext then

setRowColumn(0, 3, ctrl)

else if ctrl.Control = btnLast then

setRowColumn(0, 4, ctrl)

end;

end;

procedure TRequiredFieldsFrame.setAlign(anAl: TAlign);

begin

case anAl of

alNone:

;

alTop:

begin

if parent.align = alTop then

exit

else if align = alBottom then

align := alTop

else

begin

setH(szButtonY, szButtonX);

align := alTop;

Height := szButtonY;

end;

end;

alBottom:

begin

if align = alBottom then

exit

else if align = alTop then

align := alBottom

else

begin

setH(szButtonY, szButtonX);

align := alBottom;

Height := szButtonY;

end;

end;

alLeft:

begin

if align = alLeft then

exit

else if align = alRight then

align := alLeft

else

begin

setV(szButtonY, szButtonX);

align := alLeft;

Width := pnlLmargin.Width + pnlLmargin.Width + szButtonX;

end;

end;

alRight:

begin

if align = alRight then

exit

else if align = alLeft then

align := alRight

else

begin

setV(szButtonY, szButtonX);

align := alRight;

Width := pnlLmargin.Width + pnlLmargin.Width + szButtonX;

end;

end;

alClient:

;

alCustom:

;

end;

invalidate;

end;

procedure TRequiredFieldsFrame.adjustButtonSize(aSize: Integer);

begin

adjustButtonSizeToFont(aSize, szButtonX, szButtonY, szMarginH, szMarginV);

setAlign(ReqHighlightAlign);

end;

procedure TRequiredFieldsFrame.setAlign(anAl: Integer);

begin

case anAl of

0:

setAlign(alTop);

1:

setAlign(alBottom);

2:

setAlign(alLeft);

3:

setAlign(alRight);

else

setAlign(alTop);

end;

end;

end.

Unit Methods

| **Method Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| adjustButtonSizeToFont(aSize:Integer; var X,Y,HGap,VGap:Integer) | procedure | Adjust size of the components based on given size of the font |
|  | | |

File uRequiredFieldsHighlighter.pas changes

| **Class** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **TRequiredFieldsHighlighter** | File uRequiredFieldsHighlighter.pas  Implements Highlighting navigation buttons frame | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Components on Form** | N/A | | | |

Class Methods

| **Method Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| parseTemplateFieldName(aSource:  String): String | function | Extracts the field name from the string |
| parseTemplateFieldID(aSource:String): String | function | Extracts the field ID from the string |
| getControlValue(aControl:TObject): String | function | Provides string value assigned to control |
| getRequiredFieldsNamesValues(aParent: TWinControl): TStringList | function | Returns pairs Field Name – Value for the parent control |
| HighlightControl(Ctrl: TControl; aRequired: boolean) | procedure | Highlights the control |
|  |  |  |
| HighlightControls(aRequired:Boolean) | procedure | Highlights all controls |
| parseControlFLD(aSource:String): String | function | Extracts the field address from the string |
| IsTemplateControl(aCtrl,aCtrlMain: TObject): Boolean | function | Verification if the control belongs to template |
| getTemplateFieldByControl(aControl:TWinControl): TTemplateField | function | Returns Template field by Control |
| getRequiredFieldText(aFld:TTemplateField;aDelim:String=U):String; overload | function | Returns text assigned to the required field by template field object |
| getRequiredFieldText(aCtrl:TWinControl;aDelim:String=U):String; overload | function | Returns text assigned to the required field by control assigned to the template field object |
| getControlIndex(aControl:TWinControl): Integer | function | Provides index of control in the list |
| getControlByInd(anInd:Integer): TWinControl | function | Finds control by index in the list |
| getRequiredControls: TStringList | function | Returns list of required controls |
| getRequiredFieldsNamesValues(aParent: TWinControl): TStringList | function | Returns pairs Field Name – Value for the parent control |
| getNumberOfMissingFields(aParent: TWinControl):Integer | function | Returns number of missing required fields for the template |
| getControlInfo(aControl:TWinControl): String | function | Provides debug info |
|  |  |  |

Special References

| **Special Reference Name** | **Type** | **Description** |
| --- | --- | --- |
| fRequiredControls | TStringList | Object is used to track association “Template Control – Required field” |

Class Events

No events

| **Class** | **Description of enhancement** | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **TORDateCombo** | File uRequiredFieldsHighlighter.pas  Expantion the ORDtTm.TORDateCombo class providing the new property RequiredColor | | | | | |
| **Enhancement Category** | New | Modify | | Delete | | No Change |
| **Components on Form** | N/A | | | | | |
| **Class Properties Name** | | **Type** | | **Visibility** | | **Description** | |
| RequiredColor:TColor write setRequiredColor; | | TControl | | public | | Write only property to let object be highlighted | |
| **Class Methods** | | **Procedure/Function** | | **Description** | | | |
| setRequiredColor(aColor:TColor) | | procedure | | Sets color of the TORDateCombo components | | | |
| procedure TORDateCombo.setRequiredColor(aColor: TColor);  begin  MonthCombo.Color := aColor;  DayCombo.Color := aColor;  YearEdit.Color := aColor;  end; | | | | | | | |

Unit Methods

| **Name** | **Procedure/Function** | **Description** |
| --- | --- | --- |
| clearRequiredControls | procedure | Clears association “Filed-Control” |
| AddFieldControl(aFld: TTemplateField; aControl: TWinControl;  anID: String) | procedure | Adding control to the list |
| AddParentCheckbox(anEntry: TTemplateDialogEntry;  aControl: TWinControl); | procedure | Addingcontrol to the list |

New Code added

|  |
| --- |
| **Class Class TRequiredFieldsHighlighter**  **Modified Logic (New code)** |

unit uRequiredFieldsHighlighter;

interface

uses

uTemplateFields, System.Classes, Controls, ORFn;

type

TRequiredFieldsHighlighter = class(TObject)

private

function parseTemplateFieldName(aSource: String): String;

function parseTemplateFieldID(aSource: String): String;

function getControlValue(aControl: TObject): String;

function getRequiredFieldsNamesValues(aParent: TWinControl): TStringList;

procedure HighlightControl(Ctrl: TControl; aRequired: boolean);

public

procedure HighlightControls(aRequired: boolean);

function parseControlFLD(aSource: String): String;

function IsTemplateControl(aCtrl, aCtrlMain: TObject): boolean;

function getTemplateFieldByControl(aControl: TWinControl): TTemplateField;

function getRequiredFieldText(aFld: TTemplateField; aDelim: String = U)

: String; overload;

function getRequiredFieldText(aCtrl: TWinControl; aDelim: String = U)

: String; overload;

function getControlIndex(aControl: TWinControl): Integer;

function getControlByInd(anInd: Integer): TWinControl;

function getRequiredControls: TStringList;

function getRequiredFieldNames(aParent: TWinControl;

anIndent: String): String;

function getNumberOfMissingFields(aParent: TWinControl): Integer;

{$IFDEF DEBUG}

function getControlInfo(aControl: TWinControl): String;

{$ENDIF}

end;

procedure clearRequiredControls;

procedure AddFieldControl(aFld: TTemplateField; aControl: TWinControl;

anID: String);

procedure AddParentCheckbox(anEntry: TTemplateDialogEntry;

aControl: TWinControl);

implementation

uses

fOptionsTIUTemplates, ExtCtrls, ORCtrls, ORDtTm, Vcl.Graphics, Vcl.StdCtrls,

Vcl.ComCtrls, mTemplateFieldButton, VAUtils, SysUtils, Dialogs,

uDlgComponents;

var

fRequiredControls: TStringList;

procedure clearRequiredControls;

begin

if assigned(fRequiredControls) then

fRequiredControls.Clear;

end;

procedure AddFieldControl(aFld: TTemplateField; aControl: TWinControl;

anID: String);

var

s: String;

begin

try

s := aFld.ID + U + aFld.FldName + U + IntToStr(Integer(aFld)) + U + 'FLD:' +

anID + U + 'ctrl:' + IntToStr(Integer(aControl));;

if aFld.Required then

s := s + U + '(\*R\*)';

fRequiredControls.AddObject(s, aControl);

except

on E: Exception do

ShowMessage(E.Message);

end;

end;

procedure AddParentCheckbox(anEntry: TTemplateDialogEntry;

aControl: TWinControl);

var

s: String;

begin

try

s := IntToStr(Integer(anEntry));

fRequiredControls.AddObject(s, aControl);

except

on E: Exception do

ShowMessage(E.Message);

end;

end;

// Extension of the TORDateCombo to support RequiredColor property

type

TORDateCombo=class(ORDtTm.TORDateCombo)

private

procedure setRequiredColor(aColor:TColor);

public

property RequiredColor:TColor write setRequiredColor;

end;

procedure TORDateCombo.setRequiredColor(aColor: TColor);

begin

MonthCombo.Color := aColor;

DayCombo.Color := aColor;

YearEdit.Color := aColor;

end;

/// /////////////////////////////////////////////////////////////////////////////

function TRequiredFieldsHighlighter.parseTemplateFieldName

(aSource: String): String;

begin

Result := piece(aSource, U, 2);

end;

function TRequiredFieldsHighlighter.getTemplateFieldByControl

(aControl: TWinControl): TTemplateField;

var

ind: Integer;

fldID: String;

begin

Result := nil;

ind := fRequiredControls.IndexOfObject(aControl);

if ind > -1 then

begin

fldID := parseTemplateFieldID(fRequiredControls[ind]);

Result := getTemplateField(fldID, True);

end;

end;

Procedure TRequiredFieldsHighlighter.HighlightControl(Ctrl: TControl;

aRequired: boolean);

var

Color2Use: TColor;

begin

if not assigned(Ctrl) then

Exit;

Color2Use := clWindow;

if aRequired then

begin

if Ctrl.Enabled then

Color2Use := ReqHighlightColor

else

Color2Use := ReqHighlightDisabledColor;

end

else

begin

if ((Ctrl is TEdit) or (Ctrl is TORComboBox) or (Ctrl is TORDateCombo) or

(Ctrl is TORDateBox) or (Ctrl is TRichEdit)) then

Color2Use := clWindow

else if ((Ctrl is TORCheckBox) or (Ctrl is TfraTemplateFieldButton) or

(Ctrl is TPanel)) then

Color2Use := clBtnFace;

end;

if (Ctrl is TEdit) then

TEdit(Ctrl).Color := Color2Use

else if (Ctrl is TORComboBox) then

TORComboBox(Ctrl).Color := Color2Use

else if (Ctrl is TORDateCombo) then

TORDateCombo(Ctrl).RequiredColor := Color2Use

else if (Ctrl is TCPRSDialogDateBox) then

TCPRSDialogDateBox(Ctrl).Color := Color2Use

else if (Ctrl is TORDateBox) then

TORDateBox(Ctrl).Color := Color2Use

else if (Ctrl is TRichEdit) then

TRichEdit(Ctrl).Color := Color2Use

else if (Ctrl is TORCheckBox) then

TORCheckBox(Ctrl).Color := Color2Use

else if (Ctrl is TfraTemplateFieldButton) then

begin

TfraTemplateFieldButton(Ctrl).spRequired.Pen.Color := Color2Use;

TfraTemplateFieldButton(Ctrl).spRequired.Brush.Color := Color2Use

end

else if (Ctrl is TCPRSDialogNumber) then

begin

TCPRSDialogNumber(Ctrl).Edit.Color := Color2Use;

// uncomment and provide the value in case the font should be updated

// TCPRSDialogNumber(Ctrl).Edit.Font.Color := ???;

end

// keep TPanel processing the last to avoid overwriting the child types

else if (Ctrl is TPanel) then

TPanel(Ctrl).Color := Color2Use;

end;

function TRequiredFieldsHighlighter.getControlValue(aControl: TObject): String;

begin

Result := '';

if aControl is TEdit then

Result := TEdit(aControl).Text

else if (aControl is TRadioButton) and (TRadioButton(aControl).Checked) then

Result := TRadioButton(aControl).Caption

else if (aControl is TORComboBox) then

Result := TORComboBox(aControl).Text

else if (aControl is TMemo) then

Result := TMemo(aControl).Text

else if (aControl is TRichEdit) then

Result := TRichEdit(aControl).Text

else if (aControl is TORCheckBox) and (TORCheckBox(aControl).Checked) then

Result := TORCheckBox(aControl).Caption

else if (aControl is TCheckBox) and (TCheckBox(aControl).Checked) then

Result := TCheckBox(aControl).Caption

else if (aControl is TfraTemplateFieldButton) then

Result := TfraTemplateFieldButton(aControl).ButtonText

else if (aControl is TCPRSDialogNumber) then

Result := TCPRSDialogNumber(aControl).Edit.Text

// Result := '' // testing blank values for Number fields

else

{$IFDEF DEBUG}

ShowMessage('DEBUG ONLY MESSAGE:' + CRLF + 'Control of class "' +

aControl.ClassName + '" returns blank value')

{$ENDIF}

;

{

// Un-commentt in case TRichEdit with only CRFL is considered empty

if (aControl is TRichEdit) then

if Result = CRLF then

Result := '';

}

end;

function TRequiredFieldsHighlighter.parseTemplateFieldID

(aSource: String): String;

begin

Result := piece(aSource, U, 1);

end;

function TRequiredFieldsHighlighter.parseControlFLD(aSource: String): String;

begin

Result := piece(aSource, U, 4);

end;

function TRequiredFieldsHighlighter.getRequiredFieldText(aCtrl: TWinControl;

aDelim: String = U): String;

var

FLD: String;

i: Integer;

begin

Result := '';

if not assigned(fRequiredControls) then

Exit;

i := fRequiredControls.IndexOfObject(aCtrl);

if i < 0 then

Exit;

FLD := parseControlFLD(fRequiredControls[i]);

for i := 0 to fRequiredControls.Count - 1 do

begin

if FLD <> parseControlFLD(fRequiredControls[i]) then

continue;

Result := Result + getControlValue(fRequiredControls.Objects[i]);

end;

end;

function TRequiredFieldsHighlighter.getRequiredFieldText(aFld: TTemplateField;

aDelim: String = U): String;

var

s: String;

i: Integer;

begin

Result := '';

for i := 0 to fRequiredControls.Count - 1 do

begin

if parseTemplateFieldName(fRequiredControls[i]) <> aFld.FldName then

continue;

s := getControlValue(fRequiredControls.Objects[i]);

if s <> '' then

Result := Result + s + aDelim;

end;

if Result <> '' then

Result := copy(Result, 1, Length(Result) - Length(aDelim));

end;

procedure TRequiredFieldsHighlighter.HighlightControls(aRequired: boolean);

var

i: Integer;

procedure HighlightControlInstance(Ctrl: TWinControl; aRequired: boolean);

var

FLD: TTemplateField;

begin

FLD := getTemplateFieldByControl(Ctrl);

if assigned(FLD) then

HighlightControl(Ctrl, aRequired and FLD.Required and

(getRequiredFieldText(Ctrl) = ''));

end;

begin

for i := 0 to fRequiredControls.Count - 1 do

if assigned(fRequiredControls.Objects[i]) then

HighlightControlInstance(TWinControl(fRequiredControls.Objects[i]),

aRequired);

end;

function TRequiredFieldsHighlighter.IsTemplateControl(aCtrl,

aCtrlMain: TObject): boolean;

begin

if TWinControl(aCtrl).Parent = aCtrlMain then

Result := True

else

Result := (TWinControl(aCtrl).Parent <> nil) and

IsTemplateControl(TWinControl(aCtrl).Parent, aCtrlMain);

end;

function TRequiredFieldsHighlighter.getRequiredControls: TStringList;

begin

Result := fRequiredControls;

end;

function TRequiredFieldsHighlighter.getRequiredFieldsNamesValues

(aParent: TWinControl): TStringList;

var

i, idx: Integer;

Ctrl: TWinControl;

FLD: TTemplateField;

sl: TStringList;

fldID, s: String;

begin

sl := TStringList.Create;

for i := 0 to fRequiredControls.Count - 1 do

begin

if fRequiredControls.Objects[i] is TWinControl then

begin

Ctrl := TWinControl(fRequiredControls.Objects[i]);

if not assigned(Ctrl) then

continue;

if not Ctrl.Enabled then

continue;

if IsTemplateControl(Ctrl, aParent) then

begin

FLD := getTemplateFieldByControl(Ctrl);

if not FLD.Required then

continue;

fldID := parseControlFLD(fRequiredControls[i]);

idx := sl.IndexOfName(fldID);

if idx < 0 then

sl.Add(fldID + '=' + getControlValue(Ctrl))

else

begin

s := getControlValue(Ctrl);

if s = '' then

continue;

if sl.Values[fldID] <> '' then

sl.Values[fldID] := sl.Values[fldID] + U + s

else

sl.Values[fldID] := s;

end;

end;

end;

end;

Result := sl;

end;

function TRequiredFieldsHighlighter.getRequiredFieldNames(aParent: TWinControl;

anIndent: String): String;

var

i: Integer;

sl: TStringList;

s: String;

function getFieldNameByFLD(aFld: String): String;

var

i: Integer;

begin

Result := '';

for i := 0 to fRequiredControls.Count - 1 do

begin

if parseControlFLD(fRequiredControls[i]) <> aFld then

continue;

Result := parseTemplateFieldName(fRequiredControls[i]);

break;

end;

end;

begin

Result := '';

sl := getRequiredFieldsNamesValues(aParent);

for i := 0 to sl.Count - 1 do

begin

if sl.Values[sl.Names[i]] <> '' then

continue;

s := getFieldNameByFLD(sl.Names[i]);

if s <> '' then

Result := Result + anIndent + s + CRLF;

end;

sl.Free;

end;

function TRequiredFieldsHighlighter.getNumberOfMissingFields

(aParent: TWinControl): Integer;

var

sl: TStringList;

i: Integer;

begin

Result := 0;

sl := getRequiredFieldsNamesValues(aParent);

for i := 0 to sl.Count - 1 do

if trim(sl.Values[sl.Names[i]]) = '' then

inc(Result);

sl.Free;

end;

function TRequiredFieldsHighlighter.getControlIndex

(aControl: TWinControl): Integer;

begin

if not assigned(aControl) then

Result := -1

else

Result := fRequiredControls.IndexOfObject(aControl);

end;

function TRequiredFieldsHighlighter.getControlByInd(anInd: Integer)

: TWinControl;

begin

Result := nil;

try

Result := TWinControl(fRequiredControls.Objects[anInd]);

except

end;

end;

{$IFDEF DEBUG}

function TRequiredFieldsHighlighter.getControlInfo

(aControl: TWinControl): String;

var

ind: Integer;

begin

if not assigned(aControl) then

Result := 'Control is not assigned'

else

begin

ind := fRequiredControls.IndexOfObject(aControl);

if ind > -1 then

Result := Format('%s', [fRequiredControls[ind]])

else

Result := 'Field is not assigned';

end;

end;

{$ENDIF}

initialization

fRequiredControls := TStringList.Create;

finalization

fRequiredControls.Free;

end.

File rOptions.pas changes

| **Unit** | **Description of enhancement** | | | |
| --- | --- | --- | --- | --- |
| **rOptions** | File rOptions.pas  Modifications to implement Highlighting functionality (Saving/restoring user preferences to/from server) | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Methods Added** | function rpcGetSetRequiredFieldsPreferences(anAction,aData:String):String;l | | | |
| **Unit rOptions Modified Logic (New code added)** | | | | |
| // NSR20100706 AA 2015/10/08 --------------------------------------------- begin  function rpcGetSetRequiredFieldsPreferences(anAction,aData:String):String;  const  rpcName = 'ORWTIU TEMPLATE PREFERENCES';  begin  try  if anAction = 'LDPREF' then  result := sCallV(rpcName, [anAction])  else if anAction = 'SVPREF' then  result := sCallV(rpcName, [anAction,aData])  else  Result := '-2^Unsupported Acton "'+anAction+'"';  except  on E: Exception do  Result := '-3^Error executing RPC "'+E.Message+'"';  end;  end;  // NSR20100706 AA 2015/10/08 ----------------------------------------------- end | | | | |

###### CPRS Notification Alert Processing (NSR#20081008)

Summary of changes

1. The TfrmPtSel class updated to present table of user processed alerts (unit fPtSel). The TPageControl (pcProcNoti: TPageControl) object added to provide different pages for presentation of Pending and Processed alerts.
2. Pending alerts list was moved to a separate tab (“Pending” tab)
3. Processed alerts are presented on a separate tab (“Processed” tab). The panel pnlPaCanvas is used as the container for form implementing the processed alerts functionality (frmAlertsProcessed).
4. New class TfrmAlertsProcessed was created to implement functionality related to the processed alerts.
5. As the total number of processed alerts might be too big the user wants to have ability to limit the number of alert descriptions on the screen. There are 2 ways of limiting the set of alerts:

* by number of alerts
* by period of time to review

Buttons “Date Range” (cmdDateRange) and “Max # of Alerts” (cmdMaxNumber) added to implement changes of the preferences. The preference values are saved in unit variables to save them between calls to SelectPatient procedure thus the consecutive opening the patient selection dialog retains the preference values. Note that these “session” values are not saved between CPRS session – the next time the CPRS is started the preference “default” values will be taken from server. To update the “default” values use Options dialog.

1. The class TfrmOptions updated to let user change the user preferences of processed alerts table (unit fOptions).
2. Unit rOptions updated with function loading/saving user preferences of processed alerts.
3. The new dialog created for editing the user preferences of processed alerts.
4. Max number of processed alerts shown by GUI is limited to 999 (fixed by GUI). The user may set the preferred limit in Options dialog or from the Patient selection dialog.
5. By default the GUI shows alerts processed during last 7 days (The review date range is set to 7). The user may override this value in Options dialog or in Patient Selector dialog . The max number of days to review is limited by the site parameter.
6. Values of preferences (“Review date range” and “max number of records”) entered in the Options dialog are saved between CPRS sessions and are used as “user defaults”.
7. Changes to default values of preferences entered on Patient Selector Dialog are not saved between CPRS sessions

Files changes

| Unit Name | Summary of changes |
| --- | --- |
| fPtSel | Class TfrmPtSel updated to implement presentation of processed alert information |
| fAlertsProcessed | Form implements functionality related to the processed alerts presentation. Used only as a child form for TfrmPtSel |
| fOptions | Class TfrmOptions updated to implement invocation of dialog window handling parameters of processed alerts presentation |
| rOptions | Funcltion rpcGetSetProcessedAlertsPreferences added to load/store user preferences of processed alerts presentation |
| fOptionsProcessedAlerts | Class TfrmOptionsProcessedAlerts created to implement dialog for editing parameters of processed alerts presentation |
| fAlertRangeEdit | Class TfrmAlertRangeEdit created to implement dialog for editing the date range of processed alerts review |
| uConst | New message definition added |
| MinusFramed.bmp | New image to use for presentation of expanded group |
| PlusFramed.bmp | New image to use for presentation of collapsed group |

Changes by Units

Unit fPTSel

By default the indicator of group collapsed/expanded status is placed on the right of the screen and there is no property that controls position of the icon. To mimic the icon on the left the group icon is used. The icon is changed dynamically based on the group status (see lstvProcessedAlertsMouseUp, lstvProcessedAlertsClick for details)

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| *Unit TfrmPtSel* | Modified | Class TfrmPtSel  Original layout  Layout of the Patient Selection dialog prior to implementation of the NSR#20081008  Modified layout  Patient Selection Dialog layout with changes introduced by the NSR#20081008  Patient Selection Dialog Layout  Patient Selection Dialog layout with changes introduced by the NSR#20081008  Note that processed alerts table is implemented bt different fom (class TfrmAlertsProcessed) |
| pcProcNoti: TPageControl | New | Component pcProcNoti is used to separate Processed alers from Pending alerts. |
| tsPendNoti: TTabSheet | New | Component is used as the canvas for Table of pending notifications |
| tsProcessed: TTabsheet | New | Component is used as the canvas for Table of processed notifications |
| Private procedure procedure SelectPtByDFN(aDFN:String); | New | Procedure contains code from original cmdOKClick method.  The code separated from the even handler to make is possible to update the screen on selecting record from the processed alert list  procedure TfrmPtSel.SelectPtByDFN(aDFN:String);  begin// 9/23/2002: Code used to check for changed pt. DFN here, but since same patient could be  // selected twice in diff. Encounter locations, check was removed and following code runs  // no matter; in fFrame code then updates Encounter display if Encounter.Location has changed.  // NOTE: Some pieces in RPC returned arrays are modified/rearranged by ListPtByDflt call in rCore!  Patient.DFN := aDFN; // The patient object in uCore must have been created already!  Encounter.Clear;  Changes.Clear; // An earlier call to ReviewChanges should have cleared this.  if (frmPtSelOptns.SrcType = TAG\_SRC\_CLIN) and (frmPtSelOptns.cboList.ItemIEN > 0) and  IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 4)) then // Clinics, not by default.  begin  Encounter.Location := frmPtSelOptns.cboList.ItemIEN;  with cboPatient do  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 4));  end  else if (frmPtSelOptns.SrcType = TAG\_SRC\_DFLT) and (DfltPtListSrc = 'C') and  IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 4)) then  with cboPatient do // "Default" is a clinic.  begin  Encounter.Location := StrToIntDef(Piece(Items[ItemIndex], U, 10), 0); // Piece 10 is ^SC( location IEN in this case.  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 4));  end  else if ((frmPtSelOptns.SrcType = TAG\_SRC\_DFLT) and (FDfltSrc = 'Combination') and  (Copy(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 3), 1, 2) = 'Cl')) and  (IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 8))) then  with cboPatient do // "Default" combination, clinic pt.  begin  Encounter.Location := StrToIntDef(Piece(Items[ItemIndex], U, 7), 0); // Piece 7 is ^SC( location IEN in this case.  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 8));  end  else if Patient.Inpatient then // Everything else:  begin  Encounter.Inpatient := True;  Encounter.Location := Patient.Location;  Encounter.DateTime := Patient.AdmitTime;  Encounter.VisitCategory := 'H';  end;  if User.IsProvider then  Encounter.Provider := User.DUZ;  FUserCancelled := False;  FOKClicked := True;  {$IFDEF PTSEL\_HISTORY}  ptSelHistory.Insert(0,cboPatient.Text); // PaPI  {$ENDIF}  Close;  end; |
| Event handler procedure TfrmPtSel.cmdOKClick(Sender: TObject); | Modified | Code modified to isolate selection of the patient by DFN. New procedure SelectPtByDFN created. |
| Unit method SelectPatient | Modified | procedure SelectPatient(ShowNotif: Boolean; FontSize: Integer; var UserCancelled: Boolean);  Method modified to set initial value of the new TPageControl object pcProcNoti  The default page is set to 0 (Pending notifications) |
| Event handler  procedure TfrmPtSel. cmdOKClick(Sender: TObject); | Modified | The handler was modified to isolate event handler from selection of the patient in SelectPtByDFN procedure |
| Object method  procedure TfrmPtSel. SelectPtByDFN(aDFN:String); | New | Code selecting patient by DFN isolated in a procedure to avoid duplication of code for similar functionality |
| Event handler  procedure TfrmPtSel. FormCreate(Sender: TObject); | Modified | Procedure updated to add initialization of new properties  procedure TfrmAlertsProcessed.FormCreate(Sender: TObject);  begin  inherited;  DefaultButton := cmdOK;  FAlertsNotReady := False;  ShowDisabledButtonTexts;  SortedColumn := 0;  Descending := True;  paSetColumnHeaders;  setAlertByServer; // Load Processed Alerts preferences  setAlertGroupList; // Build list of possible groups  UpdateAlertInfo; // Update Processed alert description  fFocusedGroup := -1;  fFocusedItem := -1;  end; |
| Class method TfrmPtSel.ShowDisabledButtonTexts; | Modified | Procedure modified to include new components in processing of the text by the screen reader  procedure TfrmPtSel.ShowDisabledButtonTexts;  var  b: Boolean;  begin  b := ScreenReaderActive;  if b then  begin  txtCmdProcess.Visible := not cmdProcess.Enabled;  txtCmdRemove.Visible := not cmdRemove.Enabled;  txtCmdForward.Visible := not cmdForward.Enabled;  txtCmdComments.Visible := not cmdComments.Enabled;  // txtCmdDefer.Visible := not cmdDefer.Enabled;  end;  stxtDateRange.TabStop := b;  stxtMaxAlert.TabStop := b;  sTxtGroupBy.TabStop := b;  stxtBtnDateRange.Visible := b;  sTxtBtnMaxRecords.Visible := b;  end; |
| Class method procedure TfrmPtSel. AlertList; | Modified | Call of method LoadProcessedAlerts added to add loading of the processed alerts to the initial loading process |
| Procedure  procedure WMSelectPatient(var Message: TMessage); message UM\_SELECTPATIENT; | New | Procedure is used to select a patient on signal (message UM\_SELECTPATIENT) received from the frmAlertsProcessing form  procedure TfrmPtSel.WMSElectPatient(var Message: TMessage);  var  s: String;  begin  if Message.WParamLo <> 0 then  begin  s := IntToStr(Message.WParamLo);  SelectPtByDFN(s);  message.Result := 0;  end;  end; |
| Procedure  procedure TfrmPtSel.pcProcNotiResize(Sender: TObject); | New | Procedure adjusts position of the group selector on change of the form size  procedure TfrmPtSel.pcProcNotiResize(Sender: TObject);  begin  inherited;  if assigned(frmAlertsProcessed) then  frmAlertsProcessed.pnlGroupBy.Width := cmdSaveList.Width + 4;  end; |
| Procedure  procedure TfrmPtSel.pnlPtSelResize(Sender: TObject); | Modified | Procedure modified to adjust size of the component  procedure TfrmPtSel.pnlPtSelResize(Sender: TObject);  begin  frmPtSelDemog.Left := cboPatient.Left + cboPatient.Width + 9;  frmPtSelDemog.Width := pnlPtSel.Width - frmPtSelDemog.Left - 2 - cmdCancel.Width;  frmPtSelOptns.Width := cboPatient.Left - iGap;  frmPtSelOptns.orapnlMainResize(nil);  SetupDemographicsLabel;  end; |

Unit fAlertsProcessed

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit  fAlertsProcessed | New | Class implementing functionality required for processed alerts presentation |
| Class TfrmAlertsProcessed | New | Form Layout  The processed alerts form includes the table of processed alerts and buttons allowign user to control the contents of the processed alerts table (number of alerts, review period, sorting and grouping of alerts)  Processed Alerts form layout  The processed alerts form includes the table of processed alerts and buttons allowign user to control the contents of the processed alerts table (number of alerts, review period, sorting and grouping of alerts) |
| lstvProcessedAlerts: TCaptionListView | New | Component is used as the Table of pending notifications |
| cmdDateRange: TButton | New | Component initializes the change to the processed alerts Date range |
| dlgDateRange: TDateGangeDlg | New | Dialog used to change receive the preferable date range for processed alerts |
| ORAutoPanel1: TORAutpPanel | New | Component is used at the container of the components providing description of the selected date range and number of alerts found for the range |
| ORAutoPanel2: TORAutoPanel | New | Component is used as the container for buttons that control settings of user preferences. |
| cmdMaxNumber: TButton | New | Button opens dialog for setting max number of alerts the user prefer to see on the screen |
| pnlGroupBy:TPanel | New | Component is used as the container for list of available groups |
| stxtDateRange: TVA508StaticText | New | Component used to provide description of the selected date range |
| pnlRaw: TPanel | New | Panel is used as the container for debug information |
| SpeedButton1:TSpeedbutton | New | Button is used to toggle visibility of the debug components |
| stxtGroupBy: TVA508StaticText | New | List of grouping options |
| ImageList1: TImageList | New | Images used to represent collapsed/expanded statuses of the group |
| Event handler procedure cmdDateRangeClick(Sender: TObject); | New | Procedure implements selection of the new value of the date range  **procedure TfrmAlertsProcessed.cmdDateRangeClick(Sender: TObject);**  **var**  **\_min,\_max,\_Start, \_Stop: String;**  **begin**  **begin**  **\_min := FormatDateTime(fAlertRangeEdit.fmtDateTime,Now - FAlertMinDate + 1);**  **\_max := FormatDateTime(fAlertRangeEdit.fmtDateTime,Now);**  **\_start := FormatFMDateTime(fAlertRangeEdit.fmtDateTime,Round(FStrtDate));**  **\_stop := FormatFMDateTime(fAlertRangeEdit.fmtDateTime,Round(FEndDate)+ 0.23595);**  **if editAlertRange(\_Start,\_stop,\_min,\_max, fAlertMinDate)= mrOK then**  **begin**  **FStrtDate := StrToFMDateTime(\_Start);**  **FEndDate := StrToFMDateTime(\_Stop);**  **paLogDays := Round(StrDateToDate(\_Stop)) - Round(StrDateToDate(\_Start)) + 1;**  **UpdateAlertInfo; // Update description of the processed alerts**  **if ScreenReaderActive then**  **GetScreenReader.Speak('Updated Date Range starts on '+**  **FormatDateTime('mmm/dd/yyyy', FMDateTimeToDateTime(FStrtDate)) + ' ends on ' +**  **FormatDateTime('mmm/dd/yyyy', FMDateTimeToDateTime(FEndDate))**  **);**  **LoadProcessedAlerts; // Load Processed alerts data from Server**  **end;**  **end;**  **end;** |
| Event handler  procedure lstvProcessedAlertsColumnClick(Sender: TObject; Column: TListColumn); | New | Procedure provides sorting capabilities  procedure TfrmAlertsProcessed.lstvProcessedAlertsColumnClick(Sender: TObject; Column: TListColumn);  procedure UpdateHeader(aHandle:HWND; aColumn: Integer);  var  Header: HWND;  Item: THDItem;  begin  Header := ListView\_GetHeader(aHandle);  ZeroMemory(@Item, SizeOf(Item));  Item.Mask := HDI\_FORMAT;  ListViewClearSortIndicator(aHandle, FpaSortedColumn);  if Column.Index <> FpaSortedColumn then  begin  FpaDescending := False;  FpaSortedColumn := Column.Index;  end  else  FpaDescending := not FpaDescending;  // Get the new column  Header\_GetItem(Header, SortedColumn, Item);  Item.fmt := Item.fmt and not (HDF\_SORTUP or HDF\_SORTDOWN);//remove both flags  if FpaDescending then  Item.fmt := Item.fmt or HDF\_SORTDOWN//include the sort descending flag  else  Item.fmt := Item.fmt or HDF\_SORTUP;//include the sort ascending flag  Header\_SetItem(Header, FpaSortedColumn, Item);  with TListView(Sender) do  begin  SortType := stText;  Items.BeginUpdate;  AlphaSort;  Items.EndUpdate;  end;  end;  begin  UpdateHeader(TListView(Sender).Handle, FpaSortedColumn);  end; |
| Event handler  procedure lstvProcessedAlertsCompare(Sender: TObject; Item1, Item2: TListItem; Data: Integer; var Compare: Integer); | New | Procedure is used as part of the sorting algorithm.  procedure TfrmAlertsProcessed.lstvProcessedAlertsCompare(Sender: TObject; Item1, Item2: TListItem; Data: Integer; var Compare: Integer);  begin  if FpaSortedColumn = 0 then  Compare := CompareText(Item1.Caption, Item2.Caption)  else  if FpaSortedColumn <> 0 then  Compare := CompareText(Item1.SubItems[FpaSortedColumn-1], Item2.SubItems[FpaSortedColumn-1]);  if FpaDescending then  Compare := -Compare;  end; |
| Event handler  procedure TfrmAlertsProcessed.cmdMaxNumberClick | New | Procedure opens input dialog for getting maximum namber of processed alerts to be presented in the table on click of the “MaxNumber” button  **procedure TfrmAlertsProcessed.cmdMaxNumberClick(Sender: TObject);**  **var**  **sValue: String;**  **begin**  **inherited;**  **sValue := IntToStr(fMaxAlertNum);**  **if InputQuery('Processed Alerts Preferences','Enter Max # of alerts to review',sValue) then**  **begin**  **fMaxAlertNum := StrToIntDef(sValue,paLogRecordsMax);**  **paLogRecordsMax := fMaxAlertNum;**  **updateAlertInfo;**  **LoadProcessedAlerts;**  **end;**  **end;** |
| Event handler procedure TfrmAlertsProcessed.cmbGroupByChange | New | Procedure updates screen on grouping request  **procedure TfrmAlertsProcessed.cmbGroupByChange(Sender: TObject);**  **begin**  **inherited;**  **FpaGroupedColumn := cmbGroupBy.ItemIndex -1;**  **GroupRecords(FpaGroupedColumn);**  **end;** |
| Event handler  procedure lstvProcessedAlertsDblClick(Sender: TObject); | New | Procedure requests selection of the patient when the user double clicks processed alert record associated with the patient  **procedure TfrmAlertsProcessed.lstvProcessedAlertsDblClick(Sender: TObject);**  **var**  **iTag: Integer;**  **begin**  **inherited;**  **if lstvProcessedAlerts.ItemIndex < 0 then**  **exit;**  **if assigned(lstvProcessedAlerts.Items[lstvProcessedAlerts.ItemIndex].Data) then**  **begin**  **iTag := Integer(lstvProcessedAlerts.Items[lstvProcessedAlerts.ItemIndex].Data);**  **SendMessage(parentSelector,UM\_SELECTPATIENT,iTag,0);**  **end**  **else**  **MessageBeep(0);**  **end;** |
| Event handler procedure lstvProcessedAlertsChange(Sender: TObject; Item: TListItem;  Change: TItemChange); | New | Procedure provides access to the patient demographic information when the position in the processed alerts is changed and the processed alert record is associated with the patient  **procedure TfrmAlertsProcessed.lstvProcessedAlertsChange(Sender: TObject; Item: TListItem;**  **Change: TItemChange);**  **var**  **s: String;**  **begin**  **inherited;**  **SelectedItemGroupInfo(Item);**  **if assigned(Item.Data) and (Change=ctState) then**  **begin**  **s := IntToStr(Integer(Item.Data));**  **frmPtSelDemog.ShowDemog(s);**  **end**  **else**  **frmPtSelDemog.ClearIDInfo;**  **Application.ProcessMessages;**  **end;** |
| Event handler procedure lstvProcessedAlertsCustomDrawItem(Sender: TCustomListView;  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean); | New | Procedure provides different presentation of processed alerts record based on association of patient with the record  procedure TfrmAlertsProcessed.lstvProcessedAlertsCustomDrawItem(Sender: TCustomListView;  Item: TListItem; State: TCustomDrawState; var DefaultDraw: Boolean);  begin  inherited;  if assigned(Item.Data) then  lstvProcessedAlerts.Canvas.Font.Color := clHighlight  else  lstvProcessedAlerts.Canvas.Font.Color := clWindowText;  end; |
| Event handler procedure lstvProcessedAlertsKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState); | New | Procedure implements selection of the patient record when the user selects record in the processed alerts table  procedure TfrmAlertsProcessed.lstvProcessedAlertsKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  var  aGroup: Integer;  begin  inherited;  if Key = VK\_RETURN then  if ssShift in Shift then  begin  if lstvProcessedAlerts.ItemIndex >=0 then  begin  aGroup := lstvProcessedAlerts.Items[lstvProcessedAlerts.ItemIndex].GroupID;  toggleGroupStatus(aGroup);  end;  end  else  lstvProcessedAlertsDblClick(nil);  end; |
| Event handler procedure TfrmAlertsProcessed.SpeedButton1Click | New | Procedure toggles visibility of the debug panel  procedure TfrmAlertsProcessed.SpeedButton1Click(Sender: TObject);  begin  inherited;  pnlRaw.Visible := not pnlRaw.Visible;  end; |
| Procedure  TfrmAlertsProcessed.toggleGroupStatus(aGroup:Integer); | New | Implementation of group status toggling  procedure TfrmAlertsProcessed.toggleGroupStatus(aGroup:Integer);  begin  if (aGroup <0) or (aGroup>=lstvProcessedAlerts.Groups.Count) then  exit;  if lgsCollapsed in lstvProcessedAlerts.Groups[aGroup].State then  begin  lstvProcessedAlerts.Groups[aGroup].State := lstvProcessedAlerts.Groups[aGroup].State - [lgsCollapsed];  lstvProcessedAlerts.Groups[aGroup].TitleImage := 2;  end  else  begin  lstvProcessedAlerts.Groups[aGroup].TitleImage := 1;  lstvProcessedAlerts.Groups[aGroup].State := lstvProcessedAlerts.Groups[aGroup].State + [lgsCollapsed];  end;  end; |
| Event Handler  procedure lstvProcessedAlertsClick(Sender: TObject); | New | Procedure processing the double clicks on selected record  **procedure TfrmAlertsProcessed.lstvProcessedAlertsClick(Sender: TObject);**  **var**  **i: integer;**  **begin**  **inherited;**  **if lstvProcessedAlerts.GroupView then**  **begin**  **for i := 0 to lstvProcessedAlerts.Groups.Count - 1 do**  **begin**  **if lgsCollapsed in lstvProcessedAlerts.Groups[i].State then**  **lstvProcessedAlerts.Groups[i].TitleImage := 1**  **else**  **lstvProcessedAlerts.Groups[i].TitleImage := 2**  **end;**  **end;**  **end;** |
| Event handler procedure lstvProcessedAlertsMouseUp | New | Procedure toggles expanded/collapsed status of the alert group  procedure TfrmAlertsProcessed.lstvProcessedAlertsMouseUp(Sender: TObject;  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);  var  grp: Integer;  pnt: TPoint;  function getGroupIDByPoint(aPoint:TPoint): Integer;  var  i: integer;  j: integer;  rect: TRect;  begin  Result := -1;  for i := 0 to lstvProcessedAlerts.Groups.Count - 1 do  begin  rect.Top := LVGGR\_HEADER;  j := SendMessage(lstvProcessedAlerts.Handle, LVM\_GETGROUPRECT, i, DWORD(@rect));  if not PtInRect(rect,aPoint) then  continue;  Result := i;  break;  end;  end;  begin  inherited;  if lstvProcessedAlerts.GroupView then  begin  pnt.X := X;  if pnt.X > width - 20 then  exit; // no need to process right side indicator clicks  pnt.Y := Y;  grp := getGroupIDByPoint(pnt);  pnlRaw.Caption := Format('X: %d Y: %d GroupID: %d',[X,Y,grp]);  if grp >=0 then  if lgsCollapsed in lstvProcessedAlerts.Groups[grp].State then  begin  lstvProcessedAlerts.Groups[grp].State := lstvProcessedAlerts.Groups[grp].State - [lgsCollapsed];  lstvProcessedAlerts.Groups[grp].TitleImage := 2;  end  else  begin  lstvProcessedAlerts.Groups[grp].TitleImage := 1;  lstvProcessedAlerts.Groups[grp].State := lstvProcessedAlerts.Groups[grp].State + [lgsCollapsed];  end;  end;  end; |
|  |  |  |
| Private field  FpaDescending: Boolean; | New | Field is used to track sort order  FpaDescending: Boolean; |
| Private field  FpaSortedColumn: Integer; | New | Field is used to track the sorting column  FpaSortedColumn: Integer; |
| Private field  FpaGroupedColumn: Integer; | New | Field is used to track the grouping column  FpaGroupedColumn: Integer; |
| Private field  FpafFocusedGroup: Integer | New | Field is used to track the selected group |
| Private field  FpafFocusedItem:Integer | New | Field is used to track the selected group item |
| Private field  FpaAlertsFound: Integer | New | Field is used to save number of records found on server  fAlertsFound, |
| Private field  fMaxAlertNum: Integer; | New | Field is used to save max number of records to review  fMaxAlertNum: Integer; |
| Unit variable  FStrtDate: TFMDateTime; | New | Field keeps beginning of the date range for review of processed alerts    FStrtDate: TFMDateTime; |
| Unit variable FEndDate: TFMDateTIme; | New | Field keeps ending of the date range for review of processed alerts  FEndDate: TFMDateTIme; |
| Unit variable  FAleertMinDate:Integer | New | Variable keeps value of date between calls of SelectPatient function |
| Unit variable  FMaxAlertNum: Integer | New | Variable keeps value of date between calls of SelectPatient function |
| Private procedure procedure LoadProcessedAlerts; | New | Procedure performs loading of the table the processed alerts records found on server  procedure TfrmAlertsProcessed.LoadProcessedAlerts;  var  List: TStringList;  i: Integer;  bShowMore:Boolean;  sStartDate,sEndDate,sValue,sProcessed : String;  const  fmtDateTime = 'mm/dd/yyyy@hh:nn';  procedure ProcessRecord(aText:String);  var  ID,  ind: Integer;  anItem:TListItem;  s:String;  {  Column # Piece  0 - Info 1: flag “I” – for informational alert, blank(?) otherwise  1 - Patient 2: Alert name - Patient.  2 - location 3: Location (package name?)  3 - Urgency 4: Urgency  4 - Alert Date/Time 5: Alert Date Time  5 - Message 6: Message Text  7 - Processed on 7: ---- blank  8 - ? 8: Alert Information  9 - ? 9: ?  10 - ? 10:  11 - ? 11:  12 - ? 12:  13 - ? 13:  14 - First Displayed 14:  NSR20081008 adds several pieces to the result string:  15 Date/Time Alert First Displayed  16 Date/Time Alert First Selected  17 Date/Time Alert Processed  18 Date/Time Alert Deleted  19 Recipient Type  20 Surrogate Name  21 Acting as Surrogate For (Name)  }  begin  anItem := lstvProcessedAlerts.Items.Add;  s := Piece(aText,U,1);  if s <> 'Forwarded by: ' then  begin  anItem.Caption := Piece(aText, U, 1);  for ind := 2 to DelimCount(aText, U) + 1 do  begin  sValue := Piece(aText, U, ind);  case ind of  8:begin  s := piece(piece(sValue,';',1),',',2);  ID := StrToIntDef(s,-1);  if ID > 0 then  anItem.Data := Pointer(ID); // Pt DFN if assigned to alert;  sValue := piece(sValue,';',3);  sValue := FormatFMDateTimeStr(fmtDateTime,sValue);  end;  9..19: continue; // comment if all pieces are needed.  // uncomment if all pieces are needed. Also check paSetColumnHeaders below  // 15,16,17,18: sValue := FormatFMDateTimeStr(fmtDateTime,sValue);  end;  anItem.SubItems.Add(sValue);  end;  {$IFDEF DEBUG\_AA}  anItem.SubItems.Add(aText); // RAW value added  {$ENDIF}  end  else  begin  anItem.SubItems[5] := Piece(aText, U, 2);  s := Piece(aText, U, 3);  if Length(s) > 0 then  anItem.SubItems[8] := 'Fwd Comment: ' + s;  end;  end;  begin  Screen.Cursor := crHourGlass;  try  // PII 9/22/2015  lstvProcessedAlerts.Items.BeginUpdate;  lstvProcessedAlerts.Items.Clear;  List := TStringList.Create;  try  List.Clear;  lstvProcessedAlerts.GroupView := false;  //Load the list here  sStartDate := FloatToStr(FStrtDate);  sEndDate := FloatToStr(FEndDate);  sProcessed := '1';  {$IFDEF DEBUG\_AA}  if not cbProcessedOnly.Checked then  sProcessed := '0';  {$ENDIF}  LoadProcessedNotifications(List,sStartDate,sEndDate,intToStr(fMaxAlertNum+1),sProcessed);  fAlertsFound := List.Count;  bShowMore := fAlertsFound > fMaxAlertNum;  if bShowMore then  fAlertsFound := fMaxAlertNum;  UpdateAlertInfo(bShowMore);  for i := 0 to fAlertsFound - 1 do  ProcessRecord(List[i]);  lstvProcessedAlerts.Items.EndUpdate;  finally  List.Free;  end;  finally  Screen.Cursor := crDefault;  end;  end; |
| Private procedure  procedure setAlertByServer; | New | Procedure loads the default values of user preferences from the server an initializes the object properties  procedure TfrmAlertsProcessed.setAlertByServer;  var  i: integer;  begin  loadProcessedAlertsInfo;  FpaAlertsFound := 0;  fMaxAlertNum := paLogRecordsMax;  fEndDate := DateTimeToFMDateTime(Now);  fStrtDate := DateTimeToFMDateTime(Now-paLogDays+1);// adding 1 to count today in  end; |
| Private procedure  procedure setAlertGroupList; | New | Procedure initializes the list of columns to be used for grouping  procedure TfrmAlertsProcessed.setAlertGroupList;  var  i: integer;  begin  FpaGroupedColumn := -1;  cmbGroupBy.Items.Clear;  cmbGroupBy.Items.Add('No Groups');  pnlGroupBy.Width := pnlGroupBy.Width - 1;  for i := 0 to lstvProcessedAlerts.Columns.Count - 1 do  cmbGroupBy.Items.Add(lstvProcessedAlerts.Columns[i].Caption);  end; |
| Private procedure procedure UpdateAlertInfo(ShowMore:Boolean=False); | New | Procedure updates screen with text summary of process alerts load results (number of records found)  procedure TfrmAlertsProcessed.updateAlertInfo(ShowMore:Boolean=False);  var  msg:String;  begin  if ShowMore then  msg := format('Last %d Notifications (More data available for selected period...)',[ FpaMaxAlertNum])  else  msg := format('Found %d Notifications',[FpaAlertsFound]);  stxtMaxAlert.Caption := msg;  Application.ProcessMessages;  stxtDateRange.Caption := {'Notifications between ' +}  FormatDateTime('mm/dd/yyyy', FMDateTimeToDateTime(FStrtDate)) + ' -- ' +  FormatDateTime('mm/dd/yyyy', FMDateTimeToDateTime(FEndDate))  + ' ' + msg  ;  stxtDateRange.Width := Canvas.TextWidth(stxtDateRange.Caption)+8;  end; |
| Private procedure procedure GroupRecords(aColumn:Integer); | New | Procedure implements grouping of the table data based on column selected for grouping  **procedure TfrmAlertsProcessed.GroupRecords(aColumn:Integer);**  **var**  **Group: TListGroup;**  **sValue,sID: String;**  **i: integer;**  **begin**  **lstvProcessedAlerts.Groups.Clear;**  **lstvProcessedAlerts.GroupView := false;**  **if aColumn >= 0 then**  **begin**  **if FpaSortedColumn <> FpaGroupedColumn then // sort if needed**  **begin**  **for I := 0 to lstvProcessedAlerts.Columns.Count - 1 do**  **ListViewClearSortIndicator(lstvProcessedAlerts.handle,i);**  **FpaSortedColumn := FpaGroupedColumn;**  **lstvProcessedAlertsColumnClick(lstvProcessedAlerts, lstvProcessedAlerts.Columns[SortedColumn]);**  **end;**  **Group := nil;**  **sValue := '\\';**  **sID := '';**  **for i := 0 to lstvProcessedAlerts.Items.Count - 1 do**  **begin**  **if FpaSortedColumn = 0 then**  **sValue :=lstvProcessedAlerts.Items[i].Caption**  **else**  **sValue :=lstvProcessedAlerts.Items[i].SubItems[FpaSortedColumn-1];**  **if sValue = '' then**  **sValue := 'no value';**  **if sValue <> sID then**  **begin**  **Group := lstvProcessedAlerts.Groups.Add;**  **Group.State := [lgsNormal, lgsCollapsible] - [lgsHidden];**  **Group.Header := Format('%s: %s',[cmbGroupBy.Text,sValue]);**  **Group.HeaderAlign := taLeftJustify; //taCenter;**  **Group.Footer := '';**  **Group.FooterAlign := taLeftJustify;**  **Group.TitleImage := 2;**  **sID := sValue;**  **end;**  **if assigned(Group) then**  **lstvProcessedAlerts.Items[i].GroupID := Group.GroupID;**  **end;**  **lstvProcessedAlerts.GroupView := true;**  **if ScreenReaderSystemActive then**  **begin**  **sValue := lstvProcessedAlerts.Columns[aColumn].Caption;**  **GetScreenReader.Speak('Grouped by '+sValue+' column');**  **end;**  **end;**  **end;** |
| Private procedure procedure TfrmAlertsProcessed.paSetColumnHeaders; | New | Procedure assigns captions to the table columns  procedure TfrmAlertsProcessed.paSetColumnHeaders;  var  i: integer;  procedure newColumn(aCaption:String;aWidth:Integer);  var  lc: TListColumn;  begin  lc := TListColumn.Create(lstvProcessedAlerts.Columns);  lc.Caption := aCaption;  lc.Width := aWidth;  end;  begin  lstvProcessedAlerts.Columns.Clear;  newColumn('Info',30);  newColumn('Patient',120);  newColumn('Location',60);  newColumn('Urgency',67);  newColumn('Alert Date/Time',110);  newColumn('Message',300);  newColumn('Forwarded By/When',60);  newColumn('Processed On',110);  // uncomment in case you need to see all pieces coming from the RPC  {  newColumn('-?-',20);  newColumn('-?-',20);  newColumn('-?-',20);  newColumn('-?-',20);  newColumn('-?-',20);  newColumn('-?-',20);  newColumn('First Displayed',20);  newColumn('First Selected',20);  newColumn('Processed',20);  newColumn('Deleted',20);  newColumn('Type',120);  }  newColumn(Processed By',120);  newColumn('Acting As',120);  {$IFDEF DEBUG\_AA}  newColumn('Raw',0);  {$ENDIF}  end; |
| Private procedure procedure SelectPtByDFN(aDFN:String); | New | Procedure contains code from original cmdOKClick method.  The code separated from the even handler to make is possible to update the screen on selecting record from the processed alert list  procedure TfrmAlertsProcessed.SelectPtByDFN(aDFN:String);  begin// 9/23/2002: Code used to check for changed pt. DFN here, but since same patient could be  // selected twice in diff. Encounter locations, check was removed and following code runs  // no matter; in fFrame code then updates Encounter display if Encounter.Location has changed.  // NOTE: Some pieces in RPC returned arrays are modified/rearranged by ListPtByDflt call in rCore!  Patient.DFN := aDFN; // The patient object in uCore must have been created already!  Encounter.Clear;  Changes.Clear; // An earlier call to ReviewChanges should have cleared this.  if (frmPtSelOptns.SrcType = TAG\_SRC\_CLIN) and (frmPtSelOptns.cboList.ItemIEN > 0) and  IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 4)) then // Clinics, not by default.  begin  Encounter.Location := frmPtSelOptns.cboList.ItemIEN;  with cboPatient do  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 4));  end  else if (frmPtSelOptns.SrcType = TAG\_SRC\_DFLT) and (DfltPtListSrc = 'C') and  IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 4)) then  with cboPatient do // "Default" is a clinic.  begin  Encounter.Location := StrToIntDef(Piece(Items[ItemIndex], U, 10), 0); // Piece 10 is ^SC( location IEN in this case.  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 4));  end  else if ((frmPtSelOptns.SrcType = TAG\_SRC\_DFLT) and (FDfltSrc = 'Combination') and  (Copy(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 3), 1, 2) = 'Cl')) and  (IsFMDateTime(Piece(cboPatient.Items[cboPatient.ItemIndex], U, 8))) then  with cboPatient do // "Default" combination, clinic pt.  begin  Encounter.Location := StrToIntDef(Piece(Items[ItemIndex], U, 7), 0); // Piece 7 is ^SC( location IEN in this case.  Encounter.DateTime := MakeFMDateTime(Piece(Items[ItemIndex], U, 8));  end  else if Patient.Inpatient then // Everything else:  begin  Encounter.Inpatient := True;  Encounter.Location := Patient.Location;  Encounter.DateTime := Patient.AdmitTime;  Encounter.VisitCategory := 'H';  end;  if User.IsProvider then  Encounter.Provider := User.DUZ;  FUserCancelled := False;  FOKClicked := True;  {$IFDEF PTSEL\_HISTORY}  ptSelHistory.Insert(0,cboPatient.Text); // PaPI  {$ENDIF}  Close;  end; |
| Private procedure procedure SelectedItemGroupInfo(Item: TListItem); | New | Procedure provides screen reader with the information on selected group.  procedure TfrmAlertsProcessed.SelectedItemGroupInfo(Item: TListItem);  var  msg: String;  grp: TListGroup;  begin  if (trim(cmbGroupBy.Text) <> 'No Groups') and ScreenReaderSystemActive then  begin  grp := lstvProcessedAlerts.Groups[Item.GroupID];  if (Item.Index = FpaFocusedItem) and (grp.GroupID = FpaFocusedGroup) then  exit;  FpaFocusedItem := Item.Index;  FpaFocusedGroup := grp.GroupID;  msg := 'Selected Message Group is '+grp.Header+'';  {$IFDEF DEBUG\_AA}  pnlRaw.Caption := msg;  {$ENDIF}  Application.ProcessMessages;  GetScreenReader.Speak(msg);  end;  end; |
| Unit method  ListViewClearSortIndicator | New | Procedure adds indicator of sorting to the column header  procedure ListViewClearSortIndicator(aHandle:HWND; aColumn: Integer);  var  Header: HWND;  Item: THDItem;  begin  Header := ListView\_GetHeader(aHandle);  ZeroMemory(@Item, SizeOf(Item));  Item.Mask := HDI\_FORMAT;  // Clear the previous arrow  Header\_GetItem(Header, aColumn, Item);  Item.fmt := Item.fmt and not (HDF\_SORTUP or HDF\_SORTDOWN);//remove both flags  Header\_SetItem(Header, aColumn, Item);  end; |
| Unit method SelectPatient | Modified | procedure SelectPatient(ShowNotif: Boolean; FontSize: Integer; var UserCancelled: Boolean);  Method modified to set initial value of the new TPageControl object pcProcNoti  The default page is set to 0 (Pending notifications) |
| Event handler  procedure TfrmAlertsProcessed.cmdOKClick(Sender: TObject); | Modified | The handler was modified to isolate event handler from selection of the patient in SelectPtByDFN procedure |
| Event handler  procedure TfrmAlertsProcessed.FormCreate(Sender: TObject); | Modified | Procedure updated to add initialization of new proprties  procedure TfrmAlertsProcessed.FormCreate(Sender: TObject);  begin  inherited;  DefaultButton := cmdOK;  FAlertsNotReady := False;  ShowDisabledButtonTexts;  FpaSortedColumn := 0;  FpaDescending := True;  paSetColumnHeaders;  if FAlertMinDate = 0 then  setAlertByServer; // Load Processed Alerts preferences  setAlertGroupList; // Build list of possible groups  UpdateAlertInfo; // Update Processed alert description  FpaFocusedGroup := -1;  FpaFocusedItem := -1;end; |
| Class method TfrmAlertsProcessed.ShowDisabledButtonTexts; | Modified | Procedure modified to include new components in processing of the text by the screen reader  procedure TfrmAlertsProcessed.ShowDisabledButtonTexts;  var  b: Boolean;  begin  b := **ScreenReaderActive;**  **if b then**  begin  txtCmdProcess.Visible := not cmdProcess.Enabled;  txtCmdRemove.Visible := not cmdRemove.Enabled;  txtCmdForward.Visible := not cmdForward.Enabled;  txtCmdComments.Visible := not cmdComments.Enabled;  // txtCmdDefer.Visible := not cmdDefer.Enabled;  end;  **stxtDateRange.TabStop := b;**  end; |
| Class method  procedure TfrmAlertsProcessed.AlertList; | Modified | Call of method **LoadProcessedAlerts** added to add loading of the processed alerts to the initial loading process |
| Unit Initialization section | New | Section added to set up the initial values of the unit variables;  initialization  FAlertMinDate := 0;  FStrtDate := 0.0;  FEndDate := 0.0; |

Unit fOptions

**TfrmOptions** - Modified to implement invocation of the Highlighting preferences dialog TfrmOptionsTIUTemplates

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit fOptions | Modified | Unit updated to include input of the user preferences: review date range and max number of records to pull from the server |
| Component | New | btnProcessedAlertsSettings: TButton |
| Event Handler  btnProcessedAlertsSettingsClick (Sender: TObject); | New | Procedure opens dialog window that allows user set preferred values of Processed alerts review: max number of the records to present and the default length of the review period (days)  procedure TfrmOptions.btnProcessedAlertsSettingsClick(Sender: TObject);  var  topsize, leftsize: integer;  b:boolean;  begin  Offset(topsize, -30, leftsize, -30);  b := (UpdateProcessedAlertsPreferences(topsize,leftsize) <> mrCancel);  FdirtyOtherStuff := FdirtyOtherStuff or b;  CheckApply;  end; |
| Unit(s) references | New | fOptionsProcessedAlerts, |
| Procedure TfrmOptions.LoadNotifications; | Modified | Procedure was updated to add info on processed alerts to the presentation table  procedure TfrmOptions.LoadNotifications;  // load Notification tab  var  processedAlertsInfo,  notifydefaults, surrogateinfo, flag, enableerase: string;  aResults: TStringList;  begin  aResults := TStringList.Create;  try  rpcGetNotifications(aResults);  LoadListView(lvwNotifications, aResults);  finally  FreeAndNil(aResults);  end;  lvwNotificationsColumnClick(lvwNotifications, lvwNotifications.Column[0]); // make sure sorted  notifydefaults := rpcGetNotificationDefaults;  flag := Piece(notifydefaults, '^', 2);  enableerase := Piece(notifydefaults, '^', 3);  chkNotificationsFlagged.Checked := flag = '1';  btnNotificationsRemove.Enabled := enableerase = '1';  surrogateinfo := rpcGetSurrogateInfo;  // btnSurrogate.Hint := surrogateinfo;  LabelSurrogate(surrogateinfo, lblNotificationsSurrogateText);  processedAlertsInfo := pieces(notifydefaults,U,4,5);  setProcessedAlertsInfo(ProcessedAlertsInfo); // save values in unit fOptionsProcessedAlerts  end; |

|  |
| --- |
| **Class TfrmOptions** Current Form |
| Layout of the "Notifications" tab of the Options Dialog prior to implementation of the NSR#20081008 |
| **Class TfrmOptions** Modified Form |
| Button “Processed Alerts Settings” added to the “Notifications” tab of the dialog.  Layout of the "Notifiacations" tab after implementation of NSR #20081008  Updated layout of the "Notifications" tab of the Options Dialog  Layout of the "Notifiacations" tab after implementation of NSR #20081008 |

Unit rOptions

**rOptions** - Funcltion rpcGetDaysBeforeAlertPurge added to load system settings of processed alerts

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit | Modified | rOptions |
| Forms added/modified |  | n/a |
| Components |  | n/a |
| Events |  | n/a |
| Properties |  | n/a |
| Method(s) | New | Funcltion rpcGetDaysBeforeAlertPurge returns # of days before the alert is purged from server  function rpcGetDaysBeforeAlertPurge:Integer;  var  sResult:String;  begin  CallVistA('ORWTPR GETARCHP',[],sResult);  Result := StrToIntDef(sResult, 30);  end; |
| Unit(s) references added |  | n/a |

Unit uConst

UM\_SELECTPATIENT= (WM\_USER + 9700); // request to select a patient based on Processed Alerts record

Constant identifies message from TfrmAlertsProcessed form

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit | Modified | uConst |
| Const  UM\_SELECTPATIENT | New | UM\_SELECTPATIENT= (WM\_USER + 9700); // request to select a ptient based on Processed Alerts record |

Unit fOptionsPocessedAlerts

**fOptionsProcessedAlerts** - Class frmOptionsProcessedAlerts created to implement dialog for editing parameters of processed alerts presentation

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit | New | fOptionsProcessedAlerts |
| Forms added/modified | New | TfrmOptionsProcessedAlerts |
| Component | New | Container for Panel1 component  pnlBottom: TPanel; |
| Component | New | Container for dialog buttons  Panel1: TPanel; |
| Component | New | Component is used to visually separate input fields from buttons  bvlBottom: TBevel; |
| Component | New | Button canceling user input  btnCancel: TButton; |
| Component | New | Button confirming user input  btnOK: TButton; |
| Component | New | Button restoring the default values of preferences  btnDefaults: TButton; |
| Component | New | Input field to enter preferred date range (days)  edDays: TEdit; |
| Component | New | Input field to enter preferred number of records to review  edMaxRecords: TEdit; |
| Component | New | Static text for presentation of the date range. Used by the screen reader  sTxtLogDays: TVA508StaticText; |
| Component | New | Static text for presentation of the number of records to review. Used by the screen reader  sTxtMaxRecords: TVA508StaticText; |
| Event Handler | New | Procedure validating the user input.  procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean); |
| Event Handler | New | Procedure processes the “Set Defaults” button click  procedure btnDefaultsClick(Sender: TObject); |
| Properties | New | Property for string representing preferences on server  property Info:String read getInfo write setInfo; |
| Method(s) added | New | private  procedure setDefaults;  procedure setInfo(aValue:String);  function getInfo:String;  procedure Init508; |
| Unit(s) references added | New | rOptions, uOptions, rCore , VAUtils ; |
| Unit variables defined in Interface section | New | var  frmOptionsProcessedAlerts: TfrmOptionsProcessedAlerts;  paLogDaysMax: Real;  paLogDays: Real;  paLogRecordsMax: Integer; |
| Unit method defined in Interface section | New | Function opens Processed Alerts Preferences dialog to allow user update values od processed alert review.  function UpdateProcessedAlertsPreferences(aTopValue,aLeftValue:Integer):Integer;  begin  setPaSiteDefaults;  frmOptionsProcessedAlerts := TfrmOptionsProcessedAlerts.Create(Application);  try  with frmOptionsProcessedAlerts do  begin  if (aTopValue < 0) or (aLeftValue < 0) then  Position := poScreenCenter  else  begin  Position := poDesigned;  Top := aTopvalue;  Left := aLeftvalue;  end;  Init508;  ResizeAnchoredFormToFont(frmOptionsProcessedAlerts);  setInfo(getProcessedAlertsInfo);  Result := ShowModal;  if Result <> mrCancel then  begin  paLogDays := StrToFloatDef(edDays.Text,paLogDays); // save new values in unit var  paLogRecordsMax := StrToIntDef(edMaxRecords.Text,paLogRecordsMax); // save new values in unit var  end;  end;  finally  frmOptionsProcessedAlerts.Release;  frmOptionsProcessedAlerts := nil;  end;  end; |
| Unit method defined in Interface section | New | Function returns string presentation of the preferences to save on server  function getProcessedAlertsInfo:String;  begin  Result := Format('%g^%d',[paLogDays,paLogRecordsMax]);  end; |
| Unit method (Interface section) | New | Procedure updates unit variables from its string presentation  procedure setProcessedAlertsInfo(anInfo:String);  begin  paLogDays := StrToFloatDef(Piece(anInfo,U,posLogDays),paLogDays);  paLogRecordsMax := StrToIntDef(Piece(anInfo,U,posRecordMax),paLogRecordsMax);  end;  ); |
| Unit method (Interface section) | New | Procedure pulls preferences string from the server and assigns values to the unit variables  procedure loadProcessedAlertsInfo;  var  sInfo:String;  begin  sInfo := rpcGetNotificationDefaults;  paLogDays := StrToFloatDef(Piece(sInfo,U,4),paLogDays);  paLogRecordsMax := StrToIntDef(Piece(sInfo,U,5),paLogRecordsMax);  end; |
| Unit method (Implementation section) |  | Procedure assigns the default values to the unit variables  procedure setPaGUIDefaults;  begin  paLogDays := defLogDays;  paLogRecordsMax := defLogRecordMax;  end; |
| Unit method (implementation section only) |  | Procedure assigns the default values to the unit variables based on server data  procedure setPaSiteDefaults;  begin  paLogDaysMax := rpcGetDaysBeforeAlertPurge; // server max for the user(?)  // UPDATE with RPC call for the case the defaults are coming from the server  paLogDays := defLogDays;  paLogRecordsMax := defLogRecordMax;  end; |
| Unit constants (implementation section only) |  | Constant defines the defaults set by GUI  const  defLogDays = 7.0;  defLogRecordMax = 100;  posLogDays = 1; // position of the LogDays value  posRecordMax = 2;// position of the LogRecordMax value |
| Form Layout | New | Processed alerts preferences dialog allows user to select the default values of processed alerts table settings - number of alerts to review and the time frame of the review. |
| procedure TfrmOptionsProcessedAlerts.setDefaults | New | Procedure initializes GUI components with the default values  procedure TfrmOptionsProcessedAlerts.setDefaults;  begin  setPaDefaults;  edDays.Text := Format('%g',[paLogDays]);  edMaxRecords.Text := Format('%d',[paLogRecordsMax]);  end; |
| procedure TfrmOptionsProcessedAlerts.setInfo(aValue:String); | New | Procedure initializes the input fields with values based on the string  procedure TfrmOptionsProcessedAlerts.setInfo(aValue:String);  begin  edDays.Text := piece(aValue,U,1);  edMaxRecords.Text := piece(aValue,U,2);  end; |
| Event Handler  procedure TfrmOptionsProcessedAlerts.btnDefaultsClick(Sender: TObject); | New | Procedure handles user click on “Defaults” button.  procedure TfrmOptionsProcessedAlerts.btnDefaultsClick(Sender: TObject);  begin  inherited;  SetDefaults;  end; |
| procedure TfrmOptionsProcessedAlerts.Init508; | New | Procedure initializes dialog form for use with the screen reader  procedure TfrmOptionsProcessedAlerts.Init508;  var  b: Boolean;  begin  b := ScreenReaderActive;  sTxtMaxRecords.TabStop := b;  sTxtLogDays.TabStop := b;  end; |
| Function  function TfrmOptionsProcessedAlerts.getInfo:String; | New | Function creates the preferences string based on the user input  function TfrmOptionsProcessedAlerts.getInfo:String;  begin  Result := edDays.Text + U + edMaxRecords.Text;  end; |
| Procedure  procedure TfrmOptionsProcessedAlerts.FormCloseQuery(Sender: TObject;  var CanClose: Boolean); |  | Procedure validates user input  procedure TfrmOptionsProcessedAlerts.FormCloseQuery(Sender: TObject;  var CanClose: Boolean);  var  f: Real;  begin  inherited;  f := StrToFloatDef(edDays.Text,paLogDays);  if (f > paLogDaysMax) or (f<0) then  begin  MessageDlg('The requested number of days '+FloatToStr(f) + CRLF +  'is out of the range of "0..'+FloatToStr(paLogDaysMax) + '" (days)' +CRLF+CRLF+  'Please update the value within the range or cancel the changes',  mtError,[mbOK],0);  CanClose := false;  end;  end; |

Unit fAlertRangeEdit

**TfrmAlertRangeEdit** - Class TfrmAlertRangeEdit created to implement dialog for editing the date range of processed alerts presentation. The dialog uses updated versions of the TORDateBox components that support range validation.

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Unit | New | fAlertRangeEdit |
| Forms added/modified | New | TfrmAlertRangeEdit |
| Form layout | New | The range selector dialog allows user select the time frame to review the alerts. |
| Component | New | Container for dialog buttond  pnlBottom: TPanel; |
| Component | New | Visual delimiter  bvlBottom: TBevel |
| Component | New | Dialog button  btnOK:TButton |
| Component | New | Dialog button  btnCancel:TButton |
| Component | New | Dialog button  btnRestore |
| Component | New | Container for dialog components  pnlTop:TPanel |
| Component | New | Selector of the range start date  ordtbStart: TORDateBox; |
| Component | New | Selector of the range stop date  ordtbStop: TORDateBox; |
| Component | New | Selector label  stxtStart: TVA508StaticText |
| Component | New | Selector label  stxtStop: TVA508StaticText |
| Component | New | Change indicator  stxtChanged: TVA508StaticText |
| Component | New | Description of the range  stxtRangeInfo: TVA508StaticText |
| Component | New | Description of the range dates  stxtRange: TVA508StaticText |
| Component | New | Disclaimer (hint)  stxtRange: TVA508StaticText |
| Private field  fStart:String | New | The value of the range start before changes were made |
| Private field  fStop:String | New | The value of the range stop before changes were made |
| Private field  fMin:String | New | The value of the range minimum date before changes were made |
| Private field  fMax:String | New | The value of the range maximum before changes were made |
| Private procedure  setValues | New | Procedure is used to set up values of the range boundaries selectors  procedure TfrmAlertRangeEdit.setValues(aStart, aStop, aMin, aMax: String);  var  dtStart, dtStop, dtMin, dtMax: TDateTime;  begin  fStart := aStart;  fStop := aStop;  fMin := aMin;  fMax := aMax;  dtStart := StrDateToDate(aStart);  dtStop := StrDateToDate(aStop);  dtMin := StrDateToDate(aMin);  dtMax := trunc(StrDateToDate(aMax)) + 0.99999;  ordtbStart.DateRange.MinDate := dtMin;  ordtbStart.DateRange.MaxDate := dtStop;  ordtbStart.DateSelected := dtStart;  ordtbStart.DateOnly := True;  ordtbStart.Text := FormatDateTime(fmtDateOnly, dtStart);  ordtbStop.DateRange.MinDate := dtStart;  ordtbStop.DateRange.MaxDate := dtMax;  ordtbStop.DateSelected := dtStop;  ordtbStop.DateOnly := True;  ordtbStop.Text := FormatDateTime(fmtDateOnly, dtStop);  stxtRange.Caption := FormatDateTime(fmtDateOnly,dtMin) + ' .. ' +  FormatDateTime(fmtDateOnly,dtMax);  end; |
| Event handler  procedure btnRestoreClick(Sender: TObject); | New | Procedure processes click of the “Restore” button  procedure TfrmAlertRangeEdit.btnRestoreClick(Sender: Tobject);  begin  inherited;  setValues(fStart, fStop, fMin, fMax);  btnRestore.Enabled := false;  stxtChanged.Visible := false;  end; |
| Event handler  procedure ordtbStartDateDialogClosed(Sender: Tobject); | New | Procedure processes the changes to the boundaries of the range  procedure TfrmAlertRangeEdit.ordtbStartDateDialogClosed(Sender: Tobject);  var  dt: TdateTime;  begin  inherited;  dt := TORDateBox(Sender).DateSelected;  if dt = 0 then  exit;  if Sender = ordtbStart then  ordtbStop.DateRange.MinDate := dt  else if Sender = ordtbStop then  ordtbStart.DateRange.MaxDate := dt;  btnRestore.Enabled :=  (ordtbStop.DateSelected <> strDateToDate(fStart)) or  (ordtbStart.DateSelected <> strDateToDate(fStop));  stxtChanged.Visible := btnRestore.Enabled;  end; |
| Event handler  procedure FormKeyDown(Sender: Tobject; var Key: Word; Shift: TshiftState); | New | Procedure allows to close the form when Esc is clicked  procedure TfrmAlertRangeEdit.FormKeyDown(Sender: Tobject; var Key: Word;  Shift: TshiftState);  begin  inherited;  if Key = VK\_ESCAPE then  ModalResult := mrCancel;  end; |
| Unit constant  fmtDateTime | New | Used as date format for dates processing/presentation  fmtDateTime = ‘mm/dd/yyyy@hh:nn’; |
| Unit constant  fmtDateOnly | New | Used as date format for dates presentation  fmtDateOnly = ‘mmm dd yyyy’; |
| Unit constant  fmtLimit | New | Used as format for hint presentation  fmtLimit = ‘Please note that range can’’t exceed %d day(s) limit set for this site.’; |
| Unit function defined in interface section  function editAlertRange(var aStart, aStop, aMin, aMax: String; Limit:Integer): Integer; | New | Function is used to update the user preferences  function editAlertRange(var aStart, aStop, aMin, aMax: String; Limit:Integer): Integer;  begin  if not assigned(frmAlertRangeEdit) then  Application.CreateForm(TfrmAlertRangeEdit, frmAlertRangeEdit);  try  ResizeAnchoredFormToFont(frmAlertRangeEdit);  with frmAlertRangeEdit do  begin  {$IFDEF DEBUG\_AA}  sbDebug.Visible := True;  {$ENDIF}  stxtRangeHint.Caption := Format(fmtLimit,[Limit]);  setValues(aStart, aStop, aMin, aMax);  Result := ShowModal;  if Result = mrOK then  begin  aStart := FormatDateTime(fmtDateTime, ordtbStart.DateSelected);  aStop := FormatDateTime(fmtDateTime, ordtbStop.DateSelected);  end;  end;  finally  frmAlertRangeEdit.Release;  frmAlertRangeEdit := nil;  end;  end; |
| Untin function defined in the implementation section  function StrDateToDate(aDate: String): TDateTime; |  | Function is used to convert the string dates to TDateTime.  function StrDateToDate(aDate: String): TDateTime;  var  dtDate, dtTime: Real;  sDate, sTime: String;  delim: Char;  begin  delim := '@';  if pos('@', aDate) = 0 then  delim := ' ';  sDate := piece(aDate, delim, 1);  sTime := piece(aDate, delim, 2);  dtDate := strToDate(sDate);  dtTime := strToTime(sTime);  Result := dtDate + dtTime;  end; |

###### Progress Notes Display Misleading Request # 20070817

Summary of changes

The “Show More” node is included in the document tree (All signed notes only). When double clicked the application pulls additional documents from the server.

Untis uDocTree and fNotes were modified to support “Show More” node functionality.

Files changes

| **Unit Name** | **Summary of changes** |
| --- | --- |
| uDocTree | Updated. Added constant indicating there are mote signed documents on server. Added tree node management routines, updated implementation of function BuildDocumentTree and procedure CreateListItemsForDocumentTree.  Added several general purpose TTreeView processing routines |
| fNotes | Update.  The form object was updated:   * To include fields to keep list of documents between server calls. * To modify procedures pulling data from server * Sorting Tree nodes and Document list   The code was updated to remove duplicated sections, restricted and cleaned up from unused comments. The source was ion an |
| rTIU | Updated. Procedure ListNotes replaced with the function ListNotesForTree. The function returns the last document is received from the server.  Procedures LoadDocumentText and LoadDetailText were updated to use CallVistA instead of CallV |
| dShared | Updated. New Icon added to the imgNotes image list. imgLblNotes component updated with the description of the new icon |
| fDCSumm | Updated. The call to “BuildDocumentTree2 routine replace with the call to BuildDocumentTree |
| fConsults | Updated. The call to “BuildDocumentTree2 routine replace with the call to BuildDocumentTree |
| uConst | New constant added IMG\_SHOWMORE – index of the “SHOW MORE” icon in the image list |

Changes by Units

Unit uDocTree

| Item | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **uDocTree** | File uDocTree.pas | | | |
| Description | Added constant indicating there are mote signed documents on server. Added tree node management routines, updated implementation of function BuildDocumentTree and procedure CreateListItemsForDocumentTree. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| TX\_MORE | New | Constant added to define text indicating the availability of additional documents in the tree.  const  TX\_MORE = 'SHOW MORE'; |
| Function  getChildCount | New | Function calculates number of the node childs  function getChildCount(aNode:TTreeNode): Integer;  var  Node: TTreeNode;  begin  Result := 0;  if aNode = nil then  exit;  Node := aNode.GetFirstChild;  while Node <> nil do  begin  Inc(Result);  Result := Result + getChildCount(Node);  Node := Node.GetNextChild(Node);  end;  end; |
| Function  getNodeByName | NEW | Function returns tree node by name  function getNodeByName(aName:String;aLevel: Integer; tv: TTreeView):TTreeNode;  var  tn: TTreeNode;  begin  tn := tv.Items.GetFirstNode;  while tn <> nil do  begin  if tn.Level = aLevel then  begin  if pos(aName,tn.Text)=1 then  break;  tn := tn.getNextSibling;  end  else if tn = nil then  break  else if tn.Level < aLevel then  tn := tn.getFirstChild  else  tn := tn.parent;  end;  Result := tn;  end; |
| Function  getExpandStatus | NEW | Function registers open nodes in the String List  function getExpandStatus(aTree:TTreeView):TStringList;  var  i: integer;  SL: TSTringList;  n: TTreeNode;  begin  SL := TStringList.Create;  if (not assigned(aTree)) or (aTree.Items.Count<1) then  exit;  for i := 0 to aTree.Items.Count - 1 do  begin  n := aTree.Items[i];  if n.Expanded then // register only expanded nodes  begin  if n.Data = nil then  SL.Add(n.Text+'=1')  else  SL.Add(PDocTreeObject(n.Data)^.DocID+'=1')  end;  end;  Result := SL;  end; |
| Procedure  setExpandStatus | NEW | Procedure restores tree expand status by String list  procedure setExpandStatus(aNodes:TTreeNodes;aStatus:TStringList);  var  sName:String;  i,iPos: integer;  begin  if not assigned(aStatus) then  exit;  for i := 0 to aNodes.Count-1 do  begin  if assigned(aNodes[i].Data) then  sName := PDocTreeObject(aNodes[i].Data)^.DocID  else  sName := aNodes[i].Text;  iPos := aStatus.IndexOfName(sName);  if iPos > -1 then  aNodes[i].Expanded := aStatus.Values[sName] = '1'  else if aNodes[i].HasChildren then  aNodes[i].Expanded := false;  end;  end; |
| Procedure  RemoveDuplicates | NEW | Procedure removes duplicate nodes from the tree  procedure RemoveDuplicates(aList: TStringList);  var  tmpList:TStringList;  begin  tmpList := TStringList.Create;  tmpList.Sorted := True;  tmpList.Duplicates := dupIgnore;  tmpList.Assign(aList);  aList.Assign(tmpList);  tmpList.Free;  end; |
| procedure  adjustOrder | NEW | Procedure moves “Show More” node to the bottom of the nodes list  procedure adjustOrder(aList: TStringList);  var  iPos,  i,j: integer;  s,ss: string;  begin  if not assigned(aList) then  exit;  iPos := -1;  for i := 0 to aList.Count - 1 do  begin  s := aList[i];  if pos(TX\_MORE,s)>0 then  begin  ss := piece(s,U,1);  j := strToIntDef(ss,-1);  if j>0 then  begin  iPos := i;  break;  end;  end;  end;  if iPos >0 then  begin  aList.Delete(iPos);  aList.Add(s);  end;  end; |
| Function  FindTreeNodeByName | NEW | Function finds the tree node by provided name based on the note StringData value  function FindTreeNodeByName(aTree:TORTreeView;aName:String):TORTreeNode;  var  i: integer;  sName:String;  begin  Result := nil;  if not assigned(aTree) then  exit;  if aTree.Items.Count <1 then  exit;  i := 0;  while (i < aTree.Items.Count) do  begin  Result := TORTreeNode(aTree.Items[i]);  sName := UpperCase(piece(Result.StringData,U,2));  if pos(aName,sName)=1 then  break  else  begin  Result := nil;  inc(i);  end;  end;  end; |
| Procedure  remapNode | NEW | Procedure adjusts position og the “Show More” node  procedure remapNode(aTree:TORTreeView;aName,aParentName:String);  var  nn,nd,ndParent:TORTreeNode;  begin  nd := findTreeNodeByName(aTree,aName);  if nd = nil then  exit;  ndParent := findTreeNodeByName(aTree,aParentName);  if ndParent = nil then  exit;  if nd.Parent = ndParent then  exit;  nn := TORTreeNode(aTree.Items.AddChildObject(ndParent,nd.Text,nd.Data));  nn.Data := nd.Data;  nn.StringData := nd.StringData;  nn.StateIndex := nd.StateIndex;  aTree.Items.Delete(nd);  nn.ImageIndex := IMG\_SHOWMORE;  nn.SelectedIndex := IMG\_SHOWMORE;  nn.StateIndex := IMG\_NONE;  end; |
| procedure  CreateListItemsForDocumentTree | Modified | Procedure was modified to remove duplicate code and remove unneeded sorting. The sorting is done by TreeView component uning OnCompare event |
| Procedure  BuildDocumentTree | Modified | Procedure was modified to implement partial load of the signed documents. The original code of the procedure was commented out |
| Procedure  BuildDocumentTree2 | Removed | Procedure was replaced with BuildDocumentTree |
| Procedure  AddListViewItem | Modified | Procedure AddListViewItem was modified to process tree node containin “Show More” indicator. |
| Procedure  SetTreeNodeImagesAndFormatting | Modified | Procedure was modified to add processing of the “SHOW MORE” nodes |

Unit rTIU

| Item | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **Unit rTIU** | File rTIU.pas | | | |
| Description | Unit contains definition of methods for communicating TIU information to anfd from VistA server | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Function  ListNotesForTree | New | Function replaces procedure ListNotesForTree  The function returns string representing the last document ID included in the result set. The result string is blank if there are no more documents on the server relevant to the request.  function ListNotesForTree(Dest: TStrings; Context: Integer; Early, Late: TFMDateTime;  Person: int64; OccLim: Integer; SortAscending: Boolean; MRFlag: Integer; IEN: String):String;  { retrieves existing progress notes for a patient according to the parameters passed in  Pieces: IEN^Title^FMDateOfNote^Patient^Author^Location^Status^Visit  Return: IEN^ExDateOfNote^Title, Location, Author^ImageCount^Visit }  var  SortSeq: Char;  i: integer;  const  SHOW\_ADDENDA = True;  begin  Result := '';  if SortAscending then SortSeq := 'A' else SortSeq := 'D';  if Context > 0 then  begin  CallVistA('TIU DOCUMENTS BY CONTEXT',  [3, Context, Patient.DFN, Early, Late, Person, OccLim, SortSeq, SHOW\_ADDENDA, 0, MRFlag, IEN],Dest);  if Dest.Count > 0 then // NSR 20070817  begin  i := -1;  if AnsiPos('SHOW MORE',Dest[Dest.Count-1]) > 0 then  i := Dest.Count - 1  else if AnsiPos('SHOW MORE',Dest[0]) > 0 then  i := 0;  if i > -1 then  Result := piece(Dest[i],U,1);  end;  end;  end; |
| Procedure  ListNotes | Removed | No references on this procedure were found so the procedure was commented out. |
| Procedure  LoadDocumentText | Modified | Procedure was modified to utilize CallVistA instead of the CallV |
| Procedure LoadDetailedText | Modified | Procedure was modified to utilize CallVistA instead of the CallV |
| Procedure  LoadDocumentText | Modified | Procedure modified to replace use of CallV with VistACall |
| Procedure  LoadDetailText | Modified | Procedure modified to replace use of CallV with VistACall |

Unit fNotes

| Item | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **Unit fNotes** | File fNotes.Pas | | | |
| Description | Unit contains definition of the TfrmNotes class – the form used for reviewing collections of TIU documents  The form was updated to support partial load of the signed notes. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Uses clause | Modified | References added:  StrUtils |
| Constants added to interface section  targetAllSignedNotes  targetSinedByDateRange | New | Constants define titles of the tree nodes used for presentation of different sets of notes  targetAllSignedNotes = 'All signed notes'; // NSR 20070817  targetSinedByDateRange = 'Signed notes by date range'; // NSR 20070817 |
| Object method  LoadNotes | Modified | Procedure was updated to accept a parameter contolling the scope of the document to be loaded  procedure LoadNotes(SignedOnly:Boolean=False); |
| Event Handler  lvNotesClick | New | Event handler. Procedure added to process clicks on “Show More” icon in the documents list  procedure TfrmNotes.lvNotesClick(Sender: TObject);  var  tn: TTreeNode;  begin  exit;  // inherited;  if lvNotes.Selected = nil then  exit;  if lvNotes.Selected.SubItems.Count < 0 then  exit;  if pos(TX\_MORE,lvNotes.Selected.SubItems[0]) = 1 then  begin  tn := getNodeByName(targetAllSignedNotes,0,tvNotes);  if not assigned(tn) then  tn := getNodeByName(targetSinedByDateRange,0,tvNotes);  if assigned(tn) then  begin  tn := tn.getFirstChild;  while assigned(tn) do  begin  if pos(TX\_MORE,tn.Text)>0 then  break  else  tn := tn.getNextSibling;  end;  if assigned(tn) then  tvNotes.Selected := tn; // selecting TX\_MORE node of the tree  // will force to load more notes  end;  end;  inherited;  end; |
| Event Handler  lvNotesKeyDown | New | Event handler. Procedure added to process keyboard clicks on “Show More” icon in the documents list  procedure TfrmNotes.lvNotesKeyDown(Sender: TObject; var Key: Word;  Shift: TShiftState);  begin  inherited;  if Key = VK\_RETURN then  begin  if lvNotes.Selected = nil then  exit;  if lvNotes.Selected.SubItems.Count < 1 then  exit;  if Pos(TX\_MORE, lvNotes.Selected.SubItems[0]) = 1 then  GetMoreDocuments(lvNotes);  end;  end; |
| Event Handler  lvNotesCompare | Modified | Modified to support “Show More” records that should be the last records regardless of the sort order  procedure TfrmNotes.lvNotesCompare(Sender: TObject; Item1, Item2: TListItem; Data: Integer; var Compare: Integer);  var  ix: Integer;  begin  if (pos(TX\_MORE,upperCase(Item1.SubItems[0])) > 0) then  Compare := 1  else  if (pos(TX\_MORE,upperCase(Item2.SubItems[0])) > 0) then  Compare := -1  else  begin  if ColumnToSort = 0 then  Compare := CompareText(Item1.Caption, Item2.Caption)  else  begin  ix := ColumnToSort - 1;  Compare := CompareText(Item1.SubItems[ix], Item2.SubItems[ix]);  end;  if not ColumnSortForward then  Compare := -Compare;  end;  end; |
| Event Handler  lvNotesDblClick | New | Supports double clicks of “Show More” records  procedure TfrmNotes.lvNotesDblClick(Sender: TObject);  begin  inherited;  if uChanging or (lvNotes.Selected = nil) then  exit;  if AnsiPos(TX\_MORE,lvNotes.Selected.SubItems[0]) > 0 then  GetMoreDocuments(lvNotes);  end; |
| Event Handler  lvNotesSelectItem | Modified | Modified to support “Show More” records  procedure TfrmNotes.lvNotesSelectItem(Sender: TObject; Item: TListItem; Selected: Boolean);  begin  if uChanging or (not Selected) then  exit;  uChanging := True;  with lvNotes do  if AnsiPos(TX\_MORE,Item.SUbItems[0])>0 then  begin  memNote.Lines.Text := 'Select another document or double click the line "'+TX\_MORE +'"';  stTitle.Caption := '';  memPCERead.Clear;  end  else  begin  StatusText('Retrieving selected progress note...');  lstNotes.SelectByID(Item.SubItems[5]);  UpdateActMenu(IntToStr(lstNotes.ItemIEN), lvNotes);  lstNotesClick(Self);  SendMessage(memNote.Handle, WM\_VSCROLL, SB\_TOP, 0);  end;  uChanging := False;  end; |
| Event Handler  tvNotesHint | New | Event handler. Procedure generates the Tree object hint based on the selected node.  procedure tvNotesHint(Sender: TObject; const Node: TTreeNode;var Hint: string); |
| Event Handler  tvNotesChange | Modified | Modified to include support of “Show More” nodes and use of DoSelect method  procedure TfrmNotes.tvNotesChange(Sender: TObject; Node: TTreeNode);  begin  if (tvNotes.Selected <> nil) and  (AnsiPos(TX\_MORE,tvNotes.Selected.Text) = 0) then  begin  DoSelect;  SendMessage(tvNotes.Handle, WM\_HSCROLL, SB\_THUMBTRACK, 0);  end  else  begin  SetLvNotesVisible(False);  stTitle.Caption := '';  memNote.Lines.Text := 'Select another document or double click the node "'+TX\_MORE +'"';  end;  end; |
| Event Handler  tvNotesChanging | New | Event Handler of the changes to the document tree. Procedure saves the ID of the selected document before processing “Show More” nodes  procedure tvNotesChanging(Sender: TObject; Node: TTreeNode;var AllowChange: Boolean); |
| Event Handler  tvNotesDblClick | New | Event handler of the tree double clicks. Pulls additional documents from server if the node “Show More” was clicked  procedure TfrmNotes.tvNotesDblClick(Sender: TObject);  begin  inherited;  if (tvNotes.Selected <> nil) and  (AnsiPos(TX\_MORE,tvNotes.Selected.Text)>0)  then  GetMoreDocuments;  end; |
| Event Handler  tvNotesCompare | New | Procedure controls sorting of the Tree nodes  procedure tvNotesCompare(Sender: TObject; Node1, Node2: TTreeNode;Data: Integer; var Compare: Integer); |
| Event Handler  tvNotesKeyDown | New | Event handler. Procedure processes keyboard clicks of the document tree  procedure TfrmNotes.tvNotesCompare(Sender: TObject; Node1, Node2: TTreeNode;  Data: Integer; var Compare: Integer);  begin  // check if the SortType is set for component  inherited;  if (pos(TX\_MORE,upperCase(Node1.Text)) > 0) then  Compare := 1  else  if (pos(TX\_MORE,upperCase(Node2.Text)) > 0) then  Compare := -1  else  begin  if Assigned(Node1.Data) and Assigned(Node2.Data) then  Compare := SortByDate(Node1,Node2,Data)  else  begin  if Node1.Level = Node2.Level then  Compare := 0  else if Node1.Level < Node2.Level then  Compare := -1  else  Compare := 1;  end;  if not FCurrentContext.TreeAscending then  Compare := - Compare;  end;  end; |
| Event Handler  lvNotesDblClick | New | Event handler. Processes double clicks of the documentslist  procedure lvNotesDblClick(Sender: TObject); |
| Object field  fLastDocumentID | New | Field used to save the ID of the last document selected before the “Show More” node was clicked  fLastDocumentID:String; |
| Object field  fAllSignedNotes | New | Field used to cache signed notes between server calls  fAllSignedNotes : TStringList |
| Object field  fAllUnSignedNotes | New | Field used to cache signed notes between server calls  fAllUnSignedNotes : TStringList |
| Object field  fAllUnCoSignedNotes | New | Field used to cache uncosigned notes between server calls  fAllUnCoSignedNotes : TStringList; |
| Object field  LastID | New | Field used to save ID of the tree node between calls to server  LastID:String; |
| Object field  slExpandStatus | New | Field used to save the expanded status of the tree nodes between the calles to the server  slExpandStatus: TStringList; |
| Object method  SelectNode | New | Procedure implements Node selection. Separated from the event handler to make it possible to execute selection outside of the component interaction  procedure SelectNode(aNode:TTreeNode=nil); |
| Object method  DoSelect; | New | Procedure encapsulates steps required for setting up screen when a tree node is selected  procedure DoSelect; |
| Object method  GetMoreDocuments | New | Procedure implements pulling documents from the server initiated by tree or the list object (Sender)  procedure GetMoreDocuments(Sender:TObject=nil); |
| Object method  SetLvNotesVisible | New | Procedure controls visibility of the Document list and adjusts correcponding Splitter  procedure SetLvNotesVisible(aValue: Boolean); |
| Object method  updateNotesCaption | New | Procedure updated text of the static text based on the selected tree node  procedure TfrmNotes.updateNotesCaption;  var  i : Integer;  lvl1notes: Integer;  tmpstr: String;  CntNodes: Boolean;  begin  lvl1notes := 0;  CntNodes := False;  for i := 0 to tvNotes.Items.Count - 1 do  begin  if (AnsiContainsText('All signed notes', tvNotes.Items[i].Text)) OR  (AnsiContainsText('Signed notes by date range', tvNotes.Items[i].Text))  then  //ADD CODE HERE TO SAVE INDEX  CntNodes := true;  if (CntNodes) AND (tvNotes.Items.Item[i].Level = 1) AND  (NOT AnsiContainsStr(tvNotes.Items.Item[i].Text, TX\_MORE)) then  inc(lvl1notes);  end;  stNotes.Caption := SetNoteTreeLabel(FCurrentContext);  tmpstr := ' ' + IntToStr(FCurrentContext.MaxDocs) + ' ';  if AnsiContainsStr(stNotes.Caption, tmpstr) then  stNotes.Caption := StringReplace(stNotes.Caption,  IntToStr(FCurrentContext.MaxDocs), IntToStr(lvl1notes),  [rfReplaceAll, rfIgnoreCase]);  if Not AnsiContainsStr(stNotes.Caption, 'Total:') then  stNotes.Caption := StringReplace(stNotes.Caption, 'Signed Notes',  'Signed Notes' + ' (Total: ' + NoteTotal + ')',  [rfReplaceAll, rfIgnoreCase]);  stNotes.hint := stNotes.Caption;  end;; |
| Object method  SaveTreePosition | New | Procedure saves position in the documents tree prior to pulling more documents from the server  procedure TfrmNotes.SaveTreePosition; //(fix 336188)  begin  if uChanging then //ignore if change is in progress  exit;  // saving the document ID along with the parent node caption  // as the same document may fall in different categories  LastID := '';  if assigned(tvNotes.Selected) then  begin  LastID := GetNodeLocation(tvNotes.Selected) + U;  if assigned(tvNotes.Selected.Data) then  LastID := LastID + PDocTreeObject(tvNotes.Selected.Data)^.DocID  else  LastID := LastID + tvNotes.Selected.Text;  end;  end;; |
| Object method  SaveExpandStatus | New | Procedure saves expanded status of the documents tree prior to pulling more documents from the server  procedure TfrmNotes.SaveExpandStatus; //(fix 336188)  begin  if assigned(slExpandStatus) and (slExpandStatus <> nil) then  slExpandStatus.Free;  slExpandStatus := getExpandStatus(tvNotes);  end; |
| Object method  RestoreTreePosition | New | Procedure restores the position in the documents tree after to pulling more documents from the server  procedure TfrmNotes.RestoreTreePosition(bStepForward:Boolean = false); //(fix 336188)  function findLastTreeID(anID:String):TTreeNode;  var  i: Integer;  sID: String;  b: Boolean;  begin  // searching for the the node containing the document with given ID  // and having the same parent node  b := uChanging;  uChanging := True;  Result := nil;  for i := 0 to tvNotes.Items.Count - 1 do  begin  Result := tvNotes.Items[i];  sID := getNodeLocation(Result) + U;  if assigned(Result.Data) then  sID := sID + PDocTreeObject(Result.Data)^.DocID;  if sID = LastID then  break;  end;  uChanging := b;  end;  var  tn: TTreeNode;  begin  tn := findLastTreeID(LastID);  if assigned(tn) then  begin  if not bStepForward then  tvNotes.Selected := tn  else  if tn.GetNextSibling <> nil then  tvNotes.Selected := tn.GetNextSibling  else  tvNotes.Selected := tn;  end;  end; |
| Object method  RestoreExpandStatus | New | Procedure restores the expanded status of the documents tree after pulling more documents from the server  procedure TfrmNotes.RestoreExpandStatus; //(fix 336188)  begin  if assigned(slExpandStatus) then  begin  setExpandStatus(tvNotes.Items,slExpandStatus);  freeAndNil(slExpandStatus);  end;  end;; |
| Object method  getNodeLocation | New | Function returns Location information based on the Tree level  function TfrmNotes.getNodeLocation(aNode:TTReeNode):String;  begin  Result := '';  if not assigned(aNode) then  exit;  case aNode.Level of  0: Result := aNode.Text;  1: Result := aNode.Parent.Text;  2: Result := aNode.Parent.Parent.Text;  end;  end; |
| Event handler  FormCreate | Modified | Modified to include creation of string lists used for caching documents between calls to server  fLastDocumentID := '';  fAllSignedNotes := TStringList.Create;  fAllUnSignedNotes := TStringList.Create;  fAllUnCoSignedNotes := TStringList.Create; |
| Procedure  InsertAddendum | Modified | Procedure modified to replacecalls updating the memNewNote:  Original code:  //Update the richedit for the copy paste (Formatting may have changed)  GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  memNewNote.Lines.Assign(FEditNote.Lines);  Replaced with:  //Update the richedit for the copy paste (Formatting may have changed)  // GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  // memNewNote.Lines.Assign(FEditNote.Lines);  LoadDocumentText(memNewNote.Lines, lstNotes.GetIEN(EditingIndex)); |
| Procedure  DoAutosave | Modified | Procedure modified to replacecalls updating the memNewNote:  Original code:  //Update the richedit for the copy paste (Formatting may have changed)  GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  memNewNote.Lines.Assign(FEditNote.Lines);  Replaced with:  //Update the richedit for the copy paste (Formatting may have changed)  // GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  // memNewNote.Lines.Assign(FEditNote.Lines);  LoadDocumentText(memNewNote.Lines, lstNotes.GetIEN(EditingIndex)); |
| Procedure  SaveEditedNote | Modified | Procedure modified to replacecalls updating the memNewNote:  Original code:  //Update the richedit for the copy paste (Formatting may have changed)  GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  memNewNote.Lines.Assign(FEditNote.Lines);  Replaced with:  //Update the richedit for the copy paste (Formatting may have changed)  // GetNoteForEdit(FEditNote, lstNotes.GetIEN(EditingIndex));  // memNewNote.Lines.Assign(FEditNote.Lines);  LoadDocumentText(memNewNote.Lines, lstNotes.GetIEN(EditingIndex)); |
| Event Handler  FormDestroy | Modified | The procedure was modified to include releasing of the string list fields added to the form definition:  fAllSignedNotes.Free;  fAllUnSignedNotes.Free;  fAllUnCoSignedNotes.Free; |
| Procedure  UpdateTreeView | Modified | The procedure was modified to replace call to BuildDocumentTree2 with the call to BuildDocumentTree |
| Function  SortByDate | Modified | Originally the function was defined as local within the event handler tvNotesExpanded  The visibility was changed to the unit level. |
|  |  |  |

Unit dShared

| Item | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| **Unit dNotes** | File dShared.Pas | | | |
| Description | Unit contains images shared btween the application modules  The unit was updated with the new “Show More” icon and its description | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |

| Item | Category | Description of change/enhancement |
| --- | --- | --- |
| Component  imgNotes | Modified | New Icon added to the list  This is an icon with a question mark in brackets that means "show more" |
| Component  imjLblNotes | Modified | The description of the “Show More” icon was added  “Double click to see more documents” |
| Component  imjLblImages | Modified | The description of the “Show More” icon was added  “Double click to see more documents” |
|  |  |  |

##### GUI Classes

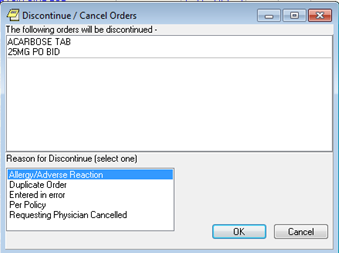
Table 38: GUI Classes

|  |  |
| --- | --- |
|  |  |
|  |  |

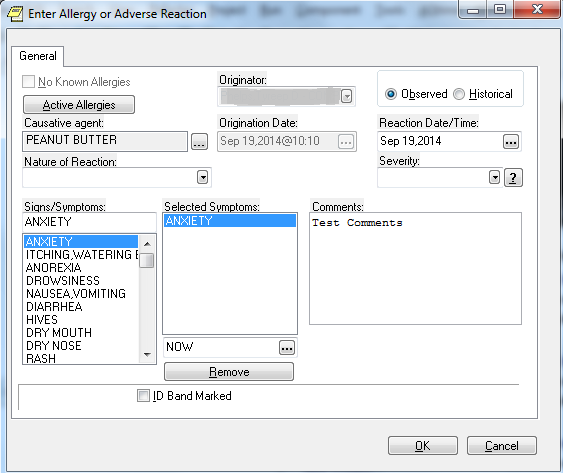
##### Current Form

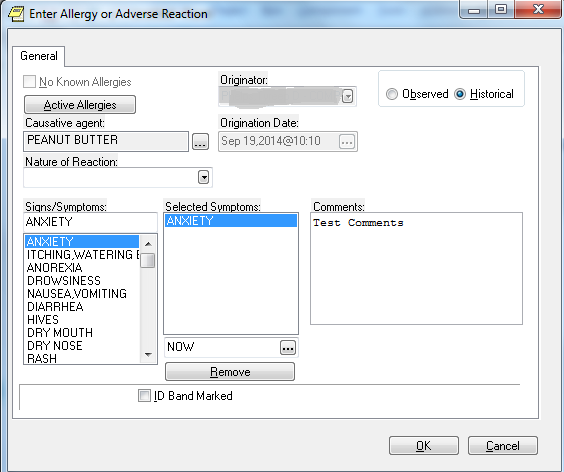
Provide a screen capture or graphical representation of the current layout.

NSR20080026

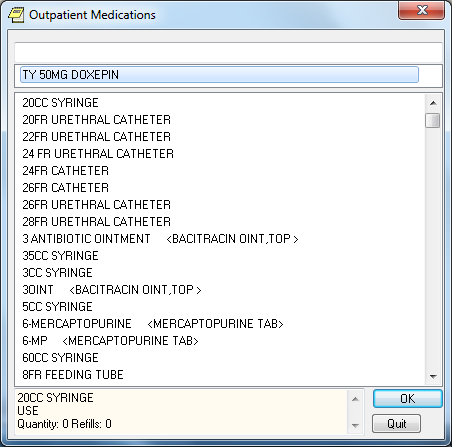


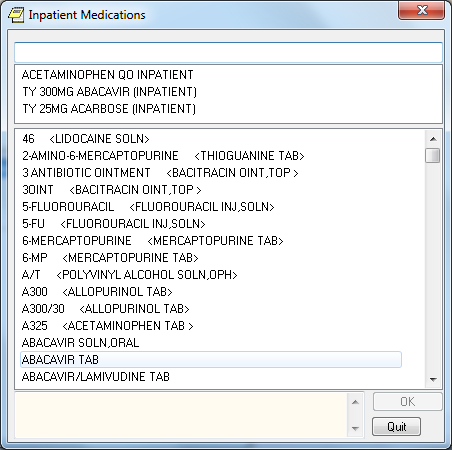
###### Adverse Reaction Reporting File Modification Request #20120404



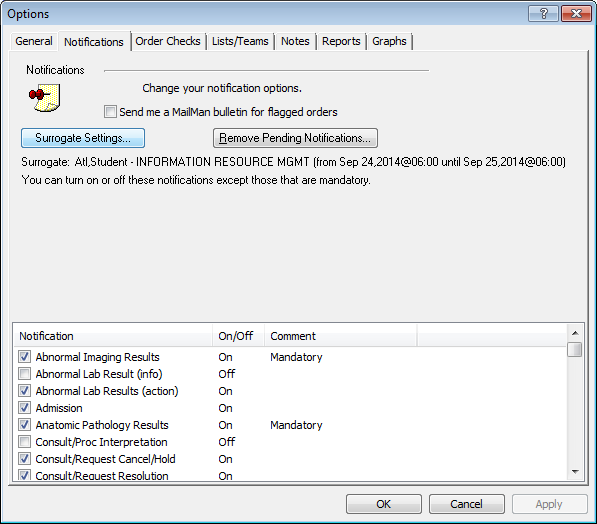


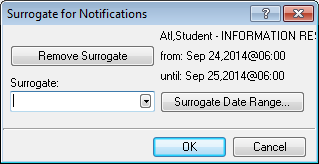
###### Drug-Allergy Order Check Enhancements and Improved Detail Request #20100825



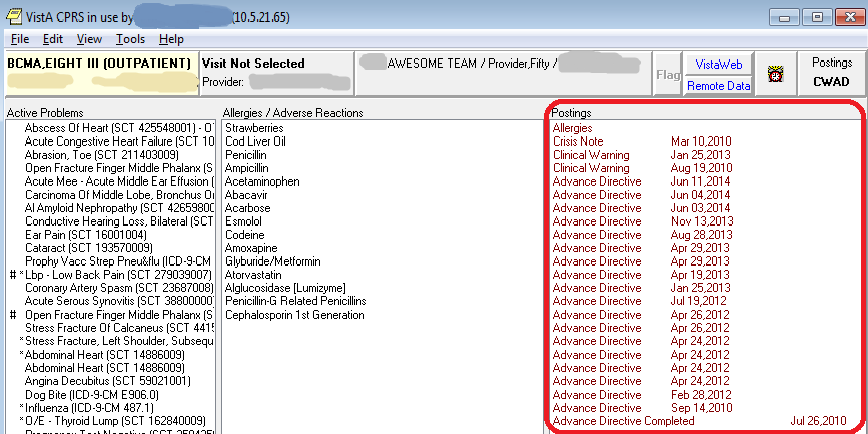


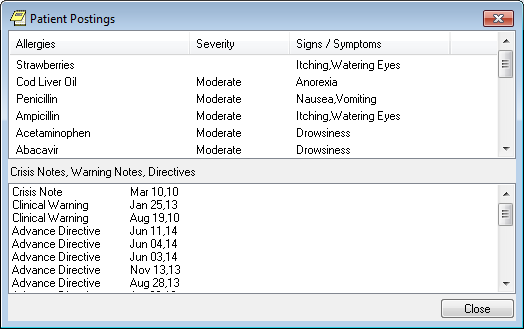
###### Update Surrogate Management Functionality within CPRS Graphical User Interface (GUI) Request #20071216



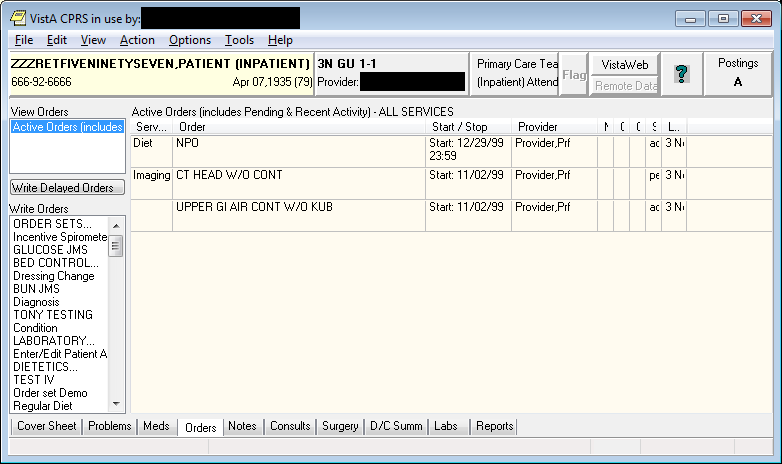


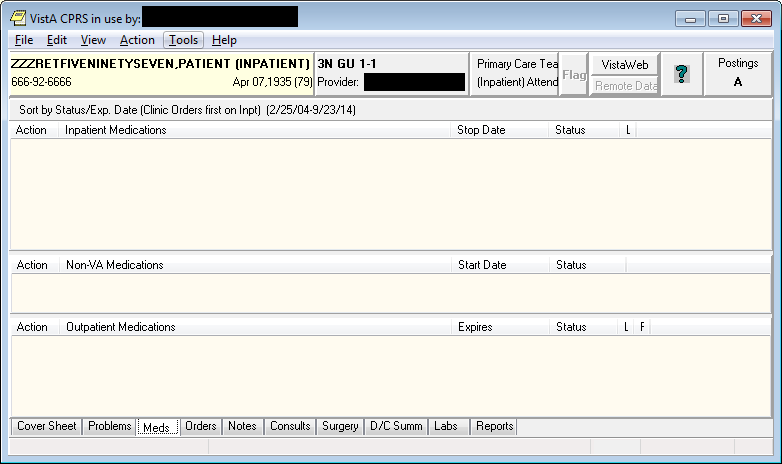
###### CWAD Post Auto-Demotion Rules Request #20080307



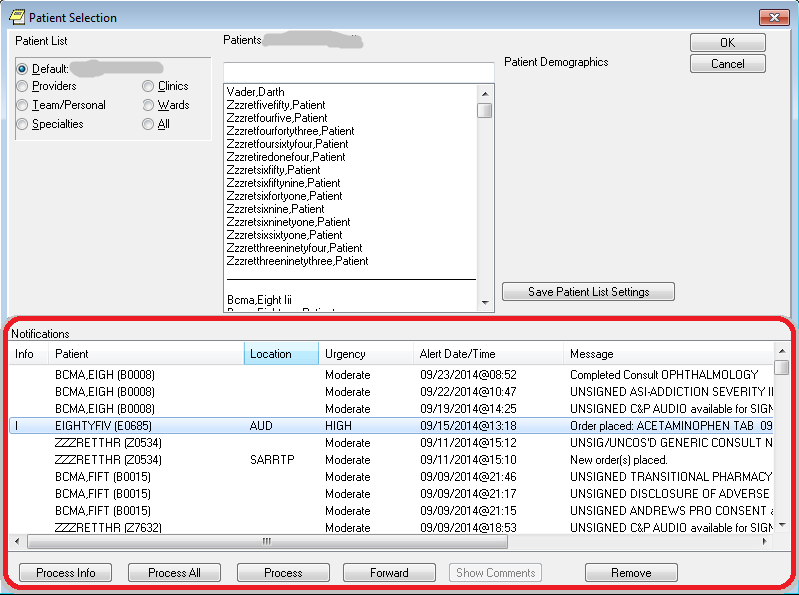


###### Prevent Confusion over CPRS Status Display of Orders and Available Actions Request #20111006

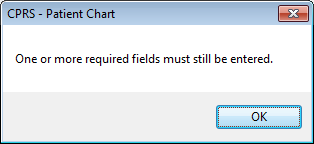


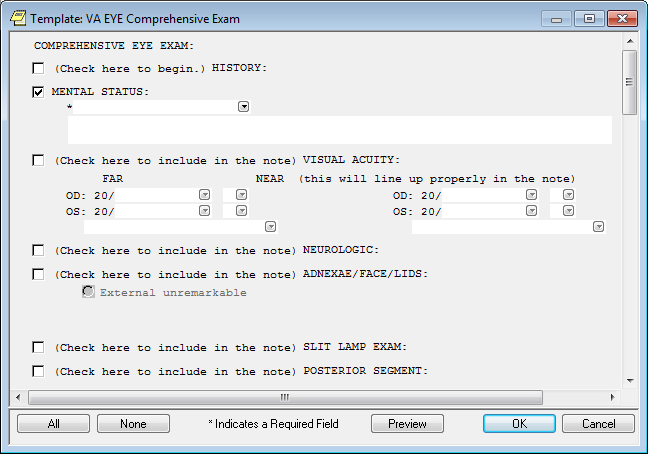


###### CPRS Notification Alert Processing Improvement Request #20081008

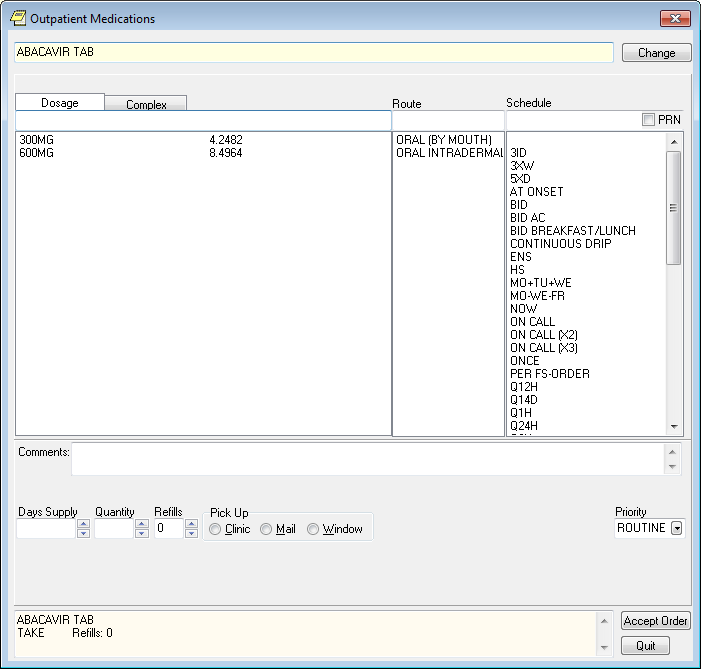


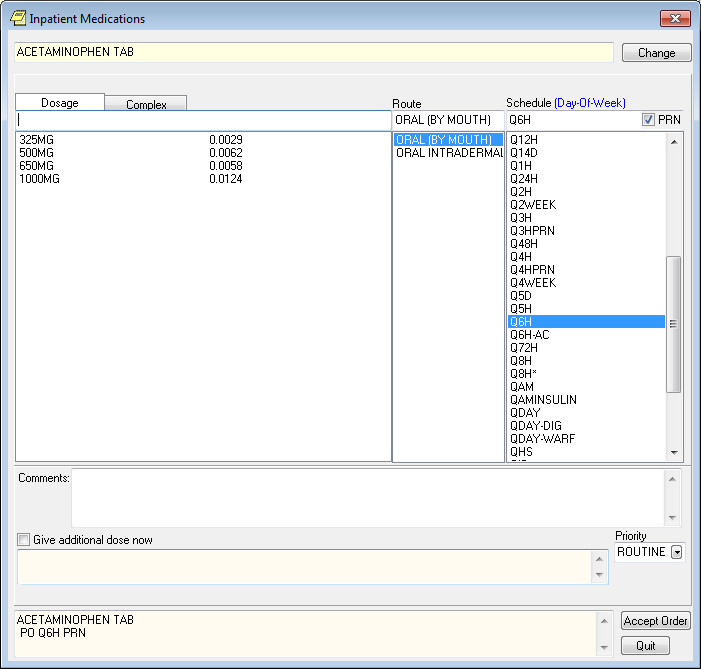
###### Identify Required Fields in TIU Note Templates Request #20100706



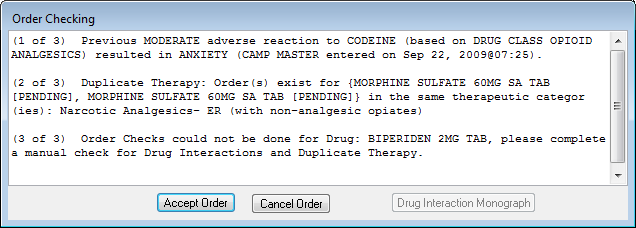


###### First Dose – Enhancement within CPRS Medication Order Request #20070811

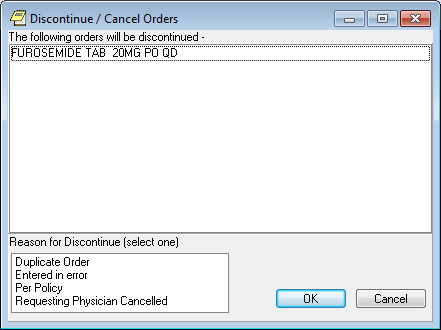




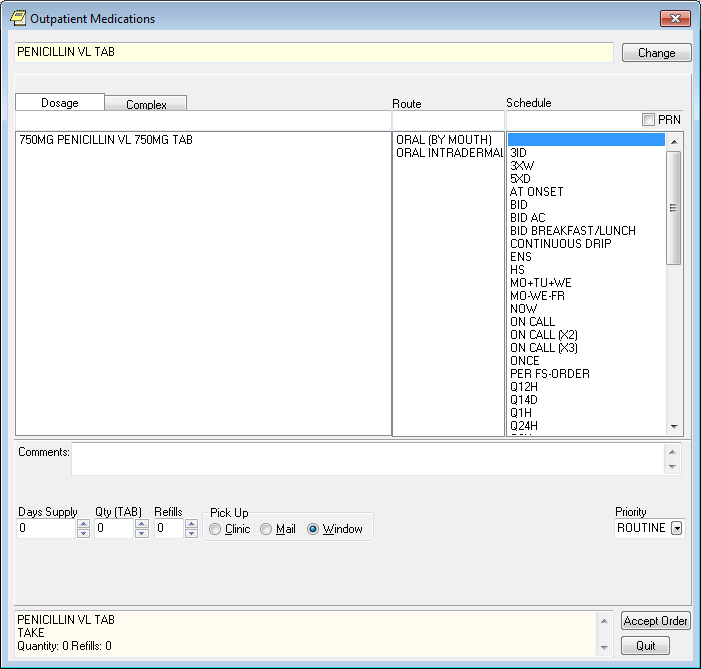
###### Critical / High Order Check Display Request #20101203

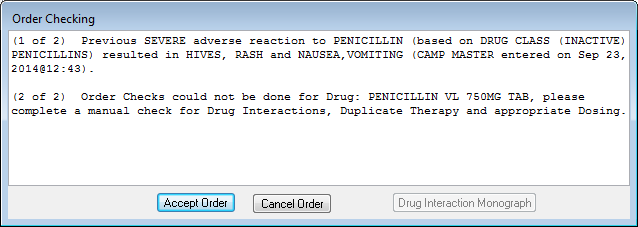


###### D/C Order by Adverse Reaction Request #20080226

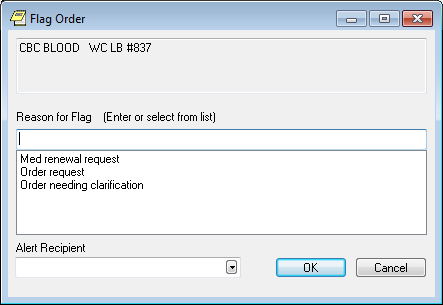


###### Changes to Allergy/Pharmacy Packages Request #20071211

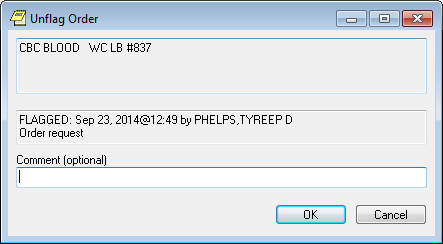




###### Order Flag Recommendations Request #20110719

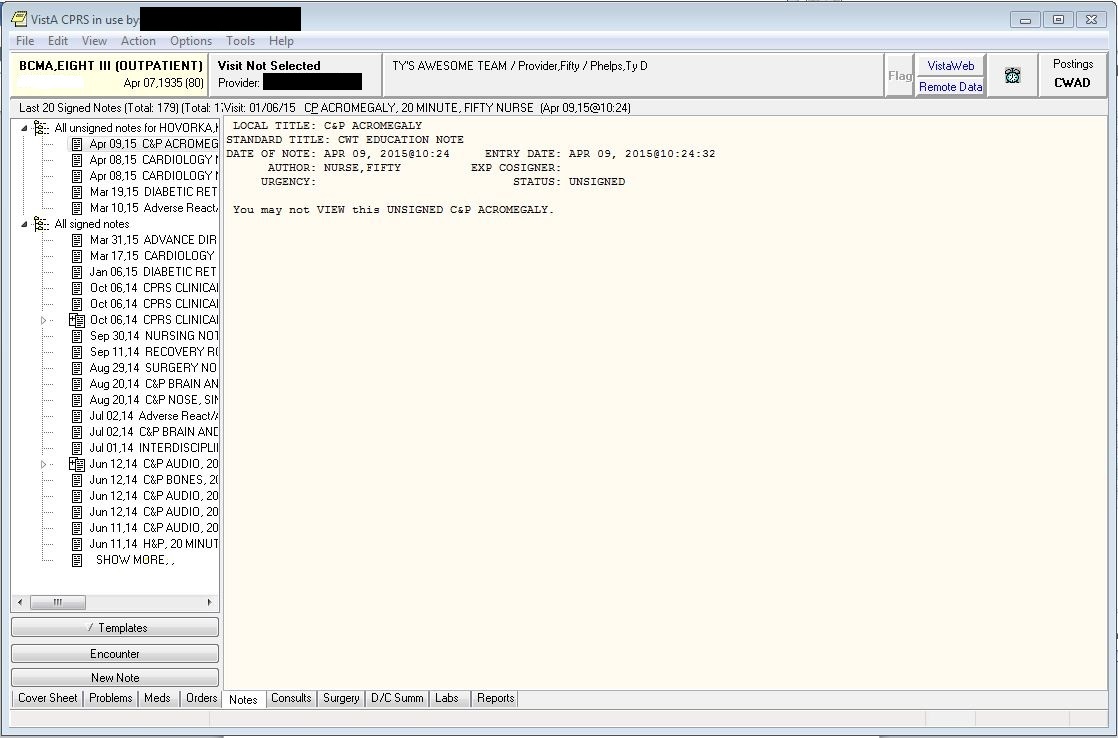


Adding Order Flag Dialog Layout



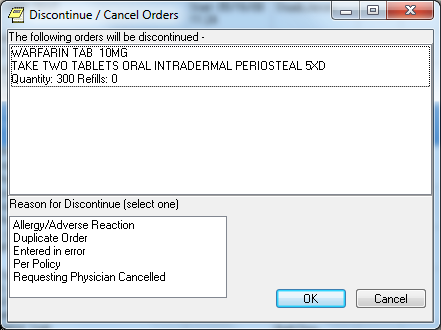
Removing Order Flag Dialog Layout

###### Progress Notes Display Misleading Request #200070817



##### Modified Form

###### D/C Order by Adverse Reaction Request #20080226



##### Components on Form

**D/C Order by Adverse Reaction Request #20080226**

Table 39: Components on Form

| Name | Type | Description |
| --- | --- | --- |
| cmdOK | Button | A button, captioned OK, to accept selections on the D/C dialog. |

##### Events

**Similar Providers #20070606**

Table 40: Events

| Name | Type | Description |
| --- | --- | --- |
| TfrmAddlSigners.btnAddSignersClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmAlertForward.btnAddAlertClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmAlertForward.cboSrcListDblClick(Sender: TObject); | OnDblClick | When double clicking on a selection in this list box this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmPtSelOptns.cboListDblClick(Sender: TObject); | OnDblClick | When double clicking on a selection in this list box this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmODCslt.ControlChange(Sender: TObject); | OnChange | OnCange this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmODProc.txtAttnChange(Sender: TObject); | OnChange | OnChange this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmDCSummProperties.cboAuthorChange(Sender: TObject); | OnChange | OnChange this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmDCSummProperties.cboAttendingChange(Sender: TObject); | OnChange | OnChange this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmAlertOrders.cmdOKClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmVisitType.btnAddClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmEncounter.cmdOKClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmNoteProperties.cboAuthorClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmNoteProperties.cboCosignerClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmPtSel.cmdOKClick(Sender: TObject); | OnClick | OnClick this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmPtSelOptns.cboListDblClick(Sender: TObject); | OnDblClick | On Double Click this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

| Name | Type | Description |
| --- | --- | --- |
| TfrmAlertOrders.cboAlertRecipientChange(Sender: TObject); | OnChange | OnChange this function will use the ID of the currently highlight cosigner on the form and retrieve a list (if any) of the similar cosigner names found based on lastname and the first two characters of the first name. |

**D/C Order by Adverse Reaction Request #20080226**

Table 41: Events

| Name | Type | Description |
| --- | --- | --- |
| cmdOKClick | OnClick | This activates the OK (acceptance) of the discontinue dialogue when the OK button is clicked/activated. |

##### Methods

**D/C Order by Adverse Reaction Request #20080226**

Table 41: Methods

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

##### Special References

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Events

Table 42: Class Events

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

##### Class Methods

Table 43: Class Methods

| Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

##### Class Properties

**D/C Order by Adverse Reaction Request #20080226**

Table 44: Class Properties

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| DCReasonTxt | String | Private | Discontinue Reason Text |

##### Uses Clause

**D/C Order by Adverse Reaction Request #20080226**

fOrdersDC

fArtAllgy

**NSR 20110606 (SIMILAR PROVIDER NAMES)**

fAddlSigners, fAlertForward, fEncnt, fNoteProps, fOrdersAlert, fPtSelOptns, fVisitType, fDCSummProps, fODConsult

fDupPro

##### Forms

Table 46: Forms

| Forms | Description | | | |
| --- | --- | --- | --- | --- |
| **Form Name** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Form Functionality** |  | | | |

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

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

##### Functions

**D/C Order by Adverse Reaction Request #20080226**

Table 48: Forms

| Function | Activities | | | |
| --- | --- | --- | --- | --- |
| **Function Name** | ExecuteDCOrders | | | |
| **Short Description** | This function executes the discontinue of an order. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** |  | | | |

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

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

| Current Logic |
| --- |
|  |
|  |
|  |
|  |

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

Table 46: Forms (List of Allergies Currently Recoded)

| Forms | Instructions |
| --- | --- |
| **Form Name** | frmDCOrdersAllrgsCrrnt |
| **Enhancement Category** | NEW |
| **Form Functionality** | 1. When Allergy/Adverse Drug Reaction is selected as reason for discontinuing a medication order, the system shall display a window presenting the user with the list of currently recorded allergies for the patient and asking whether the user wants to enter the allergy/adverse reaction. 2. If the user answers YES to proceed with Allergy/ADR entry, the system shall display the Allergy/ADR entry window within the normal discontinue medication orders workflow. 3. If the user answers NO to not proceed with Allergy/ADR entry, the system shall re-display the Orders tab, showing an unreleased order to discontinue the medication, ready for Provider signature.   The system shall include a cancel option on Allergy/ADR entry window, so the provider can elect to not enter an Allergy/ADR if not appropriate. |
| **Current Form Layout** | Allergies List. Original Layout  This is a simple rectangular shaped form that displays a list of known Historical or Obsevered allergies assigned to patient. And two buttons to confirm adding a new D/C allergy entry or cancel the action. |
| **Modified Form Layout (Changes are in bold)** | The changes to this form are: The adding of Severity and the type of reaction.  Allergies List. Modified Form Layout |

##### Functions

Table 48: List of Allergies Currently Recoded

| Functions | Instructions |
| --- | --- |
| **Function Name:** | MeasureColumnHeight |
| **Short Description** | This function is used to size the listbox and set the font. |
| **Enhancement Category** | NEW |
| **Related Options** |  |
| **Related Routines** | Function WrappedTextHeightByFont() defined in orFin.pas |
| **Data Dictionary (DD) References** |  |
| **Related Protocols** |  |
| **Related Integration Control Registrations (ICRs)** |  |
| **Data Passing** | String and Integer |
| **Input Attribute Name and Definition** | The WrappedTextHeightByFont function takes an String and an integer. The string is passed hold the description of the order to be canceled. Interger value is a place holder parameter. |
| **Output Attribute Name and Definition** | The function returns an interger result to determine if function completed successfully. |
| **Current Logic** | N/A |
| **Modified Logic (Changes are in bold)** |  |

**D/C Order by Adverse Reaction Request #20080226**

Table 48: List of Allergies Currently Recoded

| Function | Activities | | | |
| --- | --- | --- | --- | --- |
| **Function Name** | function ExecuteDCAllgryOrders(): Boolean; | | | |
| **Short Description** | This function is the initior for the creation of form class TfrmDCOrdersAllrgsCrrnt. | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Related Options** | Result of true is returned if okYesPressed button click, If not clicked false is returned.  To form TFrmOrders. | | | |

| Related Routines | Routines “Called By” | Routines “Called” |
| --- | --- | --- |
| ExecuteDCAllgryOrders | ExecuteDCOrders | procedure TfrmOrders.mnuActDCClick(Sender: TObject); |

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

| Current Logic |
| --- |
|  |

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

##### Dialog

Table 50: Dialog

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

##### Help Frame

Table 52: Help Frame

| Help Frame | Description | | | |
| --- | --- | --- | --- | --- |
| **Help Frame Text** |  | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| **Help Frame Text Calling Mechanism** |  | | | |

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

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

##### HL7 Application Parameter

Table 54: HL7 Application Parameter

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

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

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

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

##### HL7 Logical Link

Table 56: HL7 Logical Link

| HL7 Logical Link | Description |
| --- | --- |
| **HL7 Logical Link Parameter Name** |  |

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

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

**TIU Note Templates Request #20100706**

Table ??: Forms

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

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

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

| Current Logic |
| --- |
| Dialog style progress note templates can contain dialog fields that are required to be completed. When a required field is missed, the user is not directed to the specific required field that was missed. Instead, the user is now only prompted with "one or more required fields must be entered". As the user is not given direction as to which required field has been missed finding the missing required field is often very difficult.  Pt Safety: Users become frustrated and confused when a missing required field is not specifically identified by the software. This leads to cancelling out of templates and providers not entering important data in a progress note. User unfriendliness leads to providers potentially not entering important data in a progress note.   1. The user is allowed to cancel out of the note template, and then a progress note will not be created for that patient. 2. No E3Rs exist for this problem, but it has been on ongoing issue where a user begins to enter a progress note, then receives an error message about missing required fields without knowing what fields are missing. |
|  |
|  |
|  |

| Modified Logic (Changes are in bold) |
| --- |
| The new capabilities requested to satisfy the business need is to Highlight or otherwise identify which required field(s)have been missed. The Mobile Electronic Documentation software contains a feature that highlights a missing required field. The recommendation is to add this same functionality to CPRS TIU notes.  The Mobile Electronic Documentation (MED) software package was released and includes functionality that allowed missing required fields to be highlighted. Since MED and CPRS are both written in Delphi, it would seem like this same functionality would be possible in CPRS. |
|  |
|  |
|  |

Files modified

|  |  |
| --- | --- |
| fOptions.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Added button to the “Notes” tab to let user open dialog to specify preferences of ighlighting of the required fields in TIU Template. |
| fOptionsTIUTemplates.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  New Dialog implementing selector of the preferable color of highlighted field of the TIU template, location of field navigation buttons |
| uTemplateFields.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Class TTemplateDialogEntry changed to implement highlighting of the required fields without values. |
| fTemplateDialog.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Dialog implementation changed to include support of highlighting and navigation between required fields without value |
| mRequiredFields.pas | **RSD #2.612 NSR 20100706 Identify Required Fields in Text Integration Utility (TIU) Note Templates and Notify User of Missing Required Fields**  Implementation of the frame containing navigation buttons |

GUI Classes

File fOptions.pas changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| TfrmOptions | fOptions.pas  Modified to implement invocation of the Highlighting preferences dialog TfrmOptionsTIUTemplates | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| Controls added | PnlRequiredFields: TPanel;  Bevel1: TBevel;  StaticText1: TStaticText;  Memo1: TMemo;  btnNotesTitles: TButton;  imgNotes: TImage;  imgRequiredFields: TImage; | | | |
| Method(s) added | procedure btnRequiredFieldsClick(Sender: TObject); | | | |
| Unit(s) references added | fOptionsTIUTemplates, mRequiredFields; | | | |
| **Class TfrmOptions Modified Logic (New code added)** | | | | |
| procedure TfrmOptions.btnRequiredFieldsClick(Sender: TObject);  begin  restoreUserOptions;  UpdateRequiredFieldsPreferences(True);  end; | | | | |
| **Class TfrmOptions** Current Form | | | | |
| Current Layout of the Options dialog offers no control over presentation of the TIU Required fields | | | | |
| **Class TfrmOptions** Modified Form | | | | |
| The updated layout of the "Notes" tab of the "Options" dialog offers buttons for control of the presentation of the unpopulated fields within TIU templates | | | | |

File fOptionsTIUTemplates changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| TfrmTIUTemplates | File fOptionsTIUTemplates.pas  Implements Highlighting preferences dialog | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |
| Form layout | The dialog allows users to control the presentation of the unpopulated Required Fields in TIU templates - background color and position of the buttons for quick access to the unpopulated fields. | | | |
| **Class TfrmTIUTemplates Modified Logic (New code added)** | | | | |
| See files fOptionsTIUTemplates.pas; fOptionsTIUTemplates.dfm | | | | |

Components on Form

| Name | Type | Description |
| --- | --- | --- |
| btnOK | TButton | A button, captioned OK, to accept selections on values. |
| btnDefaults | TButton | Resets preferences to the ir default values |
| btnCancel | TButton | Cancels selection, restores the previous values of preferences |
| ckbHighlight | TCheckBox | Enabled/Disables the highlighting |
| rgNavigationPos | TRadioGroup | Position of the navigation bar selector |
| gbHighlightColor | TGroupBox | Group box containing the color selector |
| cbHighlightColor | TColorBox | Color selector |

Events

| Name | Type | Description |
| --- | --- | --- |
| btnDefaultsClick | OnClick | Restores default values of the preferences |
| cbHighlightColorChange | OnChange | Selects color from the drop-down list |
| rgNavigationPosClick | OnClick | Selects position of the navigation frame |
| ckbHighlightClick | OnClick | Enables/Disables the highlighting |

Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| setDefaults | procedure | Sets default values ov the preferences |
| setInfo | procedure | Info property setter |
| getInfo | function | Info property getter |
| AdjustToFontSize | procedure | Adjusts the form size based on the font size |

Special References

No changes

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Events

No changes

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Methods

No changes

| Name | Procedure/Function | Description |
| --- | --- | --- |
|  |  |  |

Class Properties

No changes

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
| Info | String | Public | String representation of the selected properties |

File uTemplateFields.pas changes

| Class | Description of enhancement | | | |
| --- | --- | --- | --- | --- |
| TTemplateDialogEntry | File uTemplateFields.pas  Modifications to implement Highlighting functionality | | | |
| **Enhancement Category** | New | Modify | Delete | No Change |

Components on Form

N/A

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Events

No changes

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Methods

| Method Name | Procedure/Function | Description |
| --- | --- | --- |
| GetControl | function | Finding control by Tag value |
| HighlightControls | procedure | Highlight controls of the template |

Special References

No changes

| Special Reference Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Class Events

No changes

| Name | Type | Description |
| --- | --- | --- |
|  |  |  |

Unit Methods

| Name | Procedure/Function | Description |
| --- | --- | --- |
| getUTmplFlds | function | Access to the uTmplFlds string list |

Class Properties

No changes

| Class Properties Name | Type | Visibility | Description |
| --- | --- | --- | --- |
|  |  |  |  |

Uses Clause

|  |  |
| --- | --- |
| Unit(s) references added to interface section | ORNet, fxBroker |
| Unit(s) references added to implementation section | fTemplateDialog, mRequiredFields; |

|  |
| --- |
| Class TTemplateDialogEntry Current Logic (function AreTemplateFieldsRequired) |
| function AreTemplateFieldsRequired(const Text: string; FldValues: TORStringList = nil): boolean;  var  flen, CtrlID, i, j: integer;  Entry: TTemplateDialogEntry;  Fld: TTemplateField;  Temp, NewTxt, FldName: string;  FoundEntry: boolean;  begin  if(not assigned(uEntries)) then  uEntries := TStringList.Create;  Temp := Text;  Result := FALSE;  repeat  i := pos(TemplateFieldBeginSignature, Temp);  if(i > 0) then  begin  CtrlID := 0;  if(copy(Temp, i + TemplateFieldSignatureLen, 1) = FieldIDDelim) then  begin  CtrlID := StrToIntDef(copy(Temp, i + TemplateFieldSignatureLen + 1, FieldIDLen-1), 0);  delete(Temp,i + TemplateFieldSignatureLen, FieldIDLen);  end;  j := pos(TemplateFieldEndSignature, copy(Temp, i + TemplateFieldSignatureLen, MaxInt));  if(j > 0) then  begin  inc(j, i + TemplateFieldSignatureLen - 1);  flen := j - i - TemplateFieldSignatureLen;  FldName := copy(Temp, i + TemplateFieldSignatureLen, flen);  Fld := GetTemplateField(FldName, FALSE);  delete(Temp,i,flen + TemplateFieldSignatureLen + 1);  end  else  begin  delete(Temp,i,TemplateFieldSignatureLen);  Fld := nil;  end;  if(CtrlID > 0) and (assigned(Fld)) and (Fld.Required) then  begin  FoundEntry := FALSE;  for j := 0 to uEntries.Count-1 do  begin  Entry := TTemplateDialogEntry(uEntries.Objects[j]);  if(assigned(Entry)) then  begin  NewTxt := Entry.GetControlText(CtrlID, TRUE, FoundEntry, FALSE);  if FoundEntry and (NewTxt = '') then{(Trim(NewTxt) = '') then //CODE ADDED BACK IN - PII}  Result := TRUE;  end;  if FoundEntry then break;  end;  if (not FoundEntry) and assigned(FldValues) then  begin  j := FldValues.IndexOfPiece(IntToStr(CtrlID));  if(j < 0) or (Piece(FldValues[j],U,2) = '') then  Result := TRUE;  end;  end;  end;  until((i = 0) or Result);  end; |

|  |
| --- |
| Class TTemplateDialogEntry Modified Logic (function AreTemplateFieldsRequired Code added is shown in **bold**) |
| function AreTemplateFieldsRequired(const Text: string; FldValues: TORStringList = nil): boolean;  var  flen, CtrlID, i, j: integer;  Entry: TTemplateDialogEntry;  Fld: TTemplateField;  Temp, NewTxt, FldName: string;  FoundEntry**, ReqFound**: boolean;  **ctrl: TControl**;  begin  if (not assigned(uEntries)) then  uEntries := TStringList.Create;  Temp := Text;  **ReqFound := FALSE;**  repeat  i := pos(TemplateFieldBeginSignature, Temp);  if(i > 0) then  begin  CtrlID := 0;  if(copy(Temp, i + TemplateFieldSignatureLen, 1) = FieldIDDelim) then  begin  CtrlID := StrToIntDef(copy(Temp, i + TemplateFieldSignatureLen + 1, FieldIDLen-1), 0);  delete(Temp,i + TemplateFieldSignatureLen, FieldIDLen);  end;  j := pos(TemplateFieldEndSignature, copy(Temp, i + TemplateFieldSignatureLen, MaxInt));  if(j > 0) then  begin  inc(j, i + TemplateFieldSignatureLen - 1);  flen := j - i - TemplateFieldSignatureLen;  FldName := copy(Temp, i + TemplateFieldSignatureLen, flen);  Fld := GetTemplateField(FldName, FALSE);  delete(Temp,i,flen + TemplateFieldSignatureLen + 1);  end  else  begin  delete(Temp,i,TemplateFieldSignatureLen);  Fld := nil;  end;  if(CtrlID > 0) and (assigned(Fld)) and (Fld.Required) then  begin  FoundEntry := FALSE;  for j := 0 to uEntries.Count-1 do  begin  Entry := TTemplateDialogEntry(uEntries.Objects[j]);  if(assigned(Entry)) then  begin  NewTxt := Entry.GetControlText(CtrlID, TRUE, FoundEntry, FALSE);  **HighlightControl(Entry.GetControl(CtrlID), FALSE); // clear any previous coloring**  **if FoundEntry and (NewTxt = '') then**  **begin**  **ctrl := Entry.GetControl(CtrlID);**  **// Mark background color**  **HighlightControl(ctrl, TRUE);**  **ReqFound := TRUE;**  end;  end;  if FoundEntry then break;  end;  if (not FoundEntry) and assigned(FldValues) then  begin  j := FldValues.IndexOfPiece(IntToStr(CtrlID));  if (j < 0) or (Piece(FldValues[j],U,2) = '') then  **ReqFound := TRUE;**  end;  end;  end;  until (i = 0);  **Result := ReqFound;**  end; |
| Class TTemplateDialogEntry Modified Logic (procedure TTemplateField.CreateDialogControls(Entry: TTemplateDialogEntry; var Index: Integer; CtrlID: integer); Code added is shown in **bold**) |
| 2 Calls to AddFieldControl procedure added to CreateDialogControls procedure  AddFieldControl(self,cb,IntToStr(ctrlID)); // NSR20100706 AA. Adding Fld.FID to track Fld by control  See Lines 1246, 1384 of uTemplateFields.pas |

|  |
| --- |
| Class TTemplateDialogEntry New Code added |
| procedure TTemplateDialogEntry.HighlightControls(aRequired:Boolean); // NSR20100706  var  ctrl: TWinControl;  i: integer;  begin  if not Assigned(fControls) then  exit;  for i := 0 to FControls.Count - 1 do  begin  ctrl := TWinControl(fControls.Objects[i]);  HighlightControlInstance(ctrl,aRequired);  end;  end;  // NSR20100706 begin  function TTemplateDialogEntry.GetControl(CtrlID: integer): TControl;  Var  I: Integer;  Ctrl: TControl;  begin  Result := nil;  for i := 0 to FControls.Count-1 do  begin  Ctrl := TControl(FControls.Objects[i]);  if(assigned(Ctrl)) and (Ctrl.Tag = CtrlID) then  Result := Ctrl;  end;  end;  // NSR20100706 end |

##### COTS Interface

Table 58: COTS Interface

| COTS Interface | Description |
| --- | --- |
| Communication Method |  |
| Application Interface |  |

## Network Detailed Design

One of the design constraints for CPRS v32 is to utilize the existing network infrastructure. Therefore no procurement is expected.

## Security and Privacy

### Security

This project will continue to use the existing VistA / CPRS security model.

### Privacy

This project will continue to use the existing VistA / CPRS privacy model.

## Service Oriented Architecture / ESS Detailed Design

CPRS is a legacy GUI application that provides a GUI front-end to the VistA system and is primarily used by physicians, nurses and other clinicians responsible for providing patient care. CPRS v32 is enhancing the existing CPRS system.

Refer to the [VistA Monograph](http://URL/VistA_Monograph.asp) for a full explanation of the larger system that CPRS v32 is a part of.

One note: CPRS’s architecture does not supply new services or consume services.

# External System Interface Design

Only one external system will be interfaced with CPRSv32. That is the interface to First DataBank, via the MOCHA server. That interface is currently nationally deployed but will be modified to include additional data to perform one additional order check. The information related to modifications to that interface are contained in the MOCHA Server SDD.

## Interface Architecture

The interface architecture for the interface to First DataBank will not be modified.

## Interface Detailed Design

Provide sufficient detail about the interface requirements for the development team to format, transmit, and/or receive data across the interface.

Include the following information (as appropriate):

* Data format requirements; if data must be reformatted before it is transmitted or after incoming data is received. Describe the tools and/or methods for the reformat process.
* Specifications for hand-shaking protocols between systems; content and format of hand-shake messages, timing for exchanging these messages, and errors handling.
* Format(s) for reports exchanged between the systems.
* Graphical representation of the connectivity between systems, showing the direction of data flow.
* Query and response descriptions.
* Describe the individual data elements that the interfacing entity(s) will provide, store, send, access, and receive, such as:
* Names/identifiers
  + Data Element Name
  + Data Format/Length
  + Data Type
  + Definition
  + Non-Technical Name
  + Non-Technical Synonyms
  + Specifications
  + Synonyms
* Range or enumeration of possible values (e.g., 0-99)
* Accuracy and precision (number of significant digits)
* Priority, timing, frequency, sequencing, and other constraints
* Security and privacy constraints
* Sources (setting/sending entities) and recipients (using/receiving entities).

Describe the data element assemblies (records, messages, files etc.) that the interfacing entity(s) will provide, store, and send, such as:

* Names/identifiers
  + Technical Name, e.g., data structure name
  + Non-technical Names, e.g. synonyms
* Data elements
* Medium/structure of data elements/assemblies
* Visual characteristics (e.g. layouts, fonts, icons etc.)
* Relationships among assemblies
* Security and privacy constraints
* Sources and recipients.

Describe the communication methods that the interfacing entity(s) will use for the interface, such as:

* Communication links, bands, frequencies, and media
* Message formatting
* Flow control (e.g. sequence numbering)
* Data transfer rate
* Routing
* Transmission services
* Safety
* Security and privacy considerations.

Describe characteristics of the protocols that the interfacing entity(s) will use for the interface, such as:

* Priority/layer of the protocol
* Packeting
* Legality checks, error control
* Recovery procedures
* Synchronization
* Status, identification, and other reporting features.

Where appropriate describe other characteristics, such as physical compatibility of the interfacing entity(s) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

# Human-Machine Interface

## Interface Design Rules

VA standards, Science Advisory & Coordinating Committee (SACC) and IBM Common User Interface (CUA).

## Inputs

Mouse and keyboard, no special or novel input device is required.

## Outputs

Please refer to section 6.2 for details regarding output changes for each of the requests included in CPRS V32.

## Navigation Hierarchy

Provide a diagram of the navigation hierarchy that shows how a user moves through the GUI.

### Screen [x.1]

Provide the layout of all input data screens or GUIs. Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary. Label each data input screen and/or GUI.

### Screen [x.2]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

### Screen [x.3]

Provide a graphic representation of each GUI, for example, a low-resolution screenshot. Define all data elements associated with each screen or GUI, or reference the data dictionary.

# 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

Attach any addition information that supplements the design specification.

* 1. Identification of Technology and Standards

Identify the system and software which apply to the SDD, including: identification number(s), title(s), abbreviation(s), version number(s), and release number(s). Identify all standards (e.g., American National Standards Institute [ANSI], International Organization for Standardization [ISO], Institute of Electrical and Electronics Engineers [IEEE], etc.).

* 1. Constraining Policies, Directives and Procedures

Identify any constraints or requirements placed on this document by policies, directives, or procedures.

* 1. Requirements Traceability Matrix

Include an RTM that traces modules and data structures to the software requirements. A reference to the location of the RTM is also acceptable.

* 1. Packaging and Installation

Outline any special considerations for software packaging and installation.

* 1. Design Metrics

Describe all metrics to be used during the design activity.

Template Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| June 2015 | 2.10 | Changed Heading 1 default setting to eliminate page break before | Process Management |
| May 2015 | 2.9 | Edited for Section 508 conformance and remediated with Common Look Office tool | Process Management |
| February 2015 | 2.8 | Incorporates revisions from PMAS Reform Lockdown; namely removing requirements for information that can be obtained from other PMAS authoritative sources. | Andrew Slawter, Office of Technology Strategies |
| September 2014 | 2.7 | Adds Enterprise Shared Services terms and requires AERB Compliance Certificate attachment. | Process Management |
| August 2014 | 2.6 | Signature block update authorized by AERB CR\_018934 | Process Management |
| March 2014 | 2.5 | Section 508 repairs to new version approved by AERB Chair approved | Process Management |
| August 2013 | 2.3 | Replaced the Service Architecture sub-section with new sub-sections for consumed and provided services. Also applied miscellaneous feedback from VA team. | ASD Enterprise Shared Services (ESS) Work Group |
| June 2013 | 1.3 | Upgraded to MS Office 2007-2010 format | Process Management |
| June 2013 | 1.2 | Address inconsistencies in Section 3, Conceptual Design, Correct headings | Process Management |
| March 2013 | 1.1 | Formatted to documentation standards and edited for Section 508 conformance | Process Management |
| January 2013 | 1.0 | Initial Document | PMAS Business Office |

Place latest revisions at top of table.

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

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

Remove blank rows.

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