Existing Product Intake Program (EPIP)

Patch OR\*3.0\*441

Remediation Plan



Department of Veterans Affairs

December 2016

Version 2.0

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 12/22/2016 | 2.0 | Updated entire document | EPIP Project Team |
| 11/28/2016 | 1.0 | Initial version | EPIP Project Team |

Table of Contents

[1. Introduction 1](#_Toc469927151)

[2. Purpose 1](#_Toc469927152)

[3. Patch Description 1](#_Toc469927153)

[3.1. Needs and Requirements 2](#_Toc469927154)

[4. Points of Contact 2](#_Toc469927155)

[5. Code Remediation 2](#_Toc469927156)

[5.1. Standards and Conventions 2](#_Toc469927157)

[5.2. Review and Analysis 3](#_Toc469927158)

[5.3. Coding Changes 3](#_Toc469927159)

[6. Testing 3](#_Toc469927160)

[6.1. Test Plan 3](#_Toc469927161)

[6.2. Test Environment 4](#_Toc469927162)

[6.3. Test Readiness Review 4](#_Toc469927163)

[6.4. Testing Phases 4](#_Toc469927164)

[6.4.1. Unit Testing 4](#_Toc469927165)

[6.4.2. Component Integration and Systems Testing (CI/ST) 4](#_Toc469927166)

[6.4.3. Functional Testing 4](#_Toc469927167)

[6.4.4. Regression Testing 5](#_Toc469927168)

[6.4.5. Section 508 Compliance Testing 5](#_Toc469927169)

[7. Documentation Remediation 5](#_Toc469927170)

[7.1. User Guides 5](#_Toc469927171)

[7.2. Installation Guides 5](#_Toc469927172)

[7.3. Technical Manuals 6](#_Toc469927173)

[7.4. Operations Manuals 6](#_Toc469927174)

[8. Project Reporting 6](#_Toc469927175)

[9. Project Schedule 6](#_Toc469927176)

[10. Deployment 6](#_Toc469927177)

[11. Sustainment Requirements 6](#_Toc469927178)

[12. Maintenance and Knowledge Transfer 6](#_Toc469927179)

[Appendix A: XINDEX Listing for MUMPS Code Changes 7](#_Toc469927180)

[Appendix B: Source Code Changes 10](#_Toc469927181)

# Introduction

The Department of Veterans Affairs (VA) currently utilizes the Veterans Health Information Systems and Technology Architecture (VistA) suite of applications to provide clinical, financial, infrastructure, and management tools. The process of advancing “Class 3” field-developed VistA software to “Class 1” nationally-distributed status is referred to as the Existing Product Intake Program (EPIP). The VA’s goal is to supplement ongoing activities associated with evaluating and advancing field-developed software to a state that meets national standards and facilitates release for Veterans Health Administration (VHA)-wide use.

# Purpose

The purpose of this document is to fully describe the remediation plan to be used for the successful remediation of the intake product code to be deployed as patch OR\*3.0\*441. This patch addresses the following NSRs:

* NSR20150103 *Display Flagged and Ward Comments*

This NSR has been implemented locally at the VA Medical Center in Milwaukee, WI.

* NSR20160613 *Anti-Microbial Decision Tree Interface Hook*

This NSR has been implemented locally at multiple Region 2 VA Medical Centers.

This document addresses the schedule, code remediation, testing, documentation, and delivery of this remediation effort.

# Patch Description

OR\*3.0\*441 provides the following enhancements to VistA:

* Modifies the display of flagged Order comments and Ward comments to make them readily available on the Computerized Patient Record System (CPRS) Orders tab where Clinical staff can plainly see Order comments and Ward comments that impact patient safety and care.

Currently, the clinician must double-click the order and search through activity details to find the flagged Order comment or Ward comment. Not only is this search time-consuming, but it requires that the clinician recognize the need to actively search. This can result in comments being overlooked and not addressed in a timely manner.

* Enables support for tracking antimicrobial medication quick orders by adding a tracking hook to enhance auditing of antimicrobial drug orders. The hook functionality is enabled by two TIU Objects provided with this enhancement: the |ZZ QUICK ORDER AUDIT| hook tracks clinically preferred antimicrobial drug orders; the |ZZ QUICK ORDER AUDIT(ALT)| hook tracks orders that prescribe an alternative to a clinically preferred antimicrobial drug.

When creating an antimicrobial drug quick order, the clinician or other authorized user must add either the |ZZ QUICK ORDER AUDIT| or the |ZZ QUICK ORDER AUDIT(ALT)| hook to the order Comments field to enable quick order audit capabilities. When an antimicrobial drug quick order containing the hook is created, the local Quick Order Audit file is populated with detailed information about the antimicrobial drug order and a Pharmacy Confirmation number is displayed on the CPRS Orders tab.

The Audit report in VistA can be used to determine how often clinicians place antimicrobial quick orders, the pharmacy confirmation number for each order, and whether the order was dispensed. Additionally, two VistA quick order auditing options are provided with this enhancement. The ORQOA QUICK ORDER AUDIT PRINT option enables printing of usage reports and export of statistics to a spreadsheet application. The ORQOA QUICK ORDER AUDIT option retrieves order details in VistA.

**Note**: Sites that are utilizing the Computer Decision Support System (CDSS) software to select quick orders associated with antimicrobial drugs can continue using the CDSS system without interruption following implementation of this enhancement.

## Needs and Requirements

The Needs and Requirements for the NSRs addressed in this remediationare:

NSR20150103 *Display Flagged and Ward Comments*

* NEED – None provided

NSR20160613 *Anti-Microbial Decision Tree Interface Hook*

* NEED 743391: Hook To Anti-Microbial Clinical Decision Support – For healthcare clinicians who utilize a Clinical Decision Support System (CDSS) to order antimicrobials for patients. The ability to track when an antimicrobial medication is ordered through the CDSS application.

# Points of Contact

The VA Point of Contact (POC) for NSR20150103 *Display Flagged and Ward Comments*

The VA POC(s) for NSR20160613 *Anti-Microbial Decision Tree Interface Hook*

# Code Remediation

Leidos will review and analyze the intake product code for compliance with coding standards, pointers, shared tables, dependencies, and any interference with VistA systems.

## Standards and Conventions

Leidos will reference the website for applicable documents and will adhere to VA standards to complete the analysis of this intake product. The output of the VA XINDEX utility will be used to analyze the MUMPS source code and document the affected routines (see Appendix A).

The MUMPS coding standards website <http://71.174.62.16/Demo/AnnoStd> will also be used to ensure that the remediated code conforms to VA standards.

## Review and Analysis

Review and analysis of this intake product involves two parts: 1) verification that the source code changes specified in this document provide the desired effect, and 2) verification that the source code changes do not adversely affect any other VistA or CPRS functionality.

Testing will be performed to validate that the intended effect of these products is implemented, and that no other VistA or CPRS Graphical User Interface (GUI) functionality is adversely affected.

## Coding Changes

The coding changes required for NSR20150103 *Display Flagged and Ward Comments* are in the following MUMPS routines:

* **Modified routines:** ORQ12
* **New routines:** None

The coding changes for NSR20160613 *Anti-Microbial Decision Tree Interface Hook* are in the following MUMPS routines:

* **Modified routines:** ORWDXM3,ORWPT
* **New routines:** ORQOAUIA, ORQOAUIB, ORQOAUIC
* **New Options:** ORQOA QUICK ANTI-MICROBIAL LST, ORQOA QUICK NIGHTLY BACKGROUND, ORQOA QUICK ORDER AUDIT, ORQOA QUICK ORDER AUDIT PRINT, ORQOA QUICK ORDER AUDIT MENU
* **New Data Objects:** ZZ QUICK ORDER AUDIT, ZZ QUICK ORDER AUDIT (ALT)

A detailed analysis of the modified and new routines included in this modification is provided in Appendix B.

# Testing

Leidos will perform all testing-related activities to ensure that the remediated code meets the expectations of the VA business owner.

## Test Plan

Leidos will configure the test environment, provide code modifications and end-to-end testing, and deliver applicable testing documentation, following Veteran-focused Integration Process (VIP) guidelines.

The Leidos developer will modify the software pursuant to the VA standards defined in the *Standards and Conventions* section of this document, and will conduct full unit testing of the functionality and verify performance of all software code before it is released to Leidos SQA. SQA will then perform all applicable testing types as described in the *Testing Phases* section of this document. The developer and SQA will resolve problems and address issues as they arise during testing and will document issues using the Rational Team Concert (RTC) defect tracking tool.

## Test Environment

Within five working days of approval of this Remediation Plan, the developer will configure the development/test environment on an Austin Information Technology Center (AITC) server or other VA-approved development/test environment used for this intake product and install the remediated Kernel Installation and Distribution System (KIDS) build. The environment will be restored to its original baseline state by the VistA system administrator after development testing is completed, followed by installation of the remediated software.

Upon notification from the developer of test environment readiness, SQA will commence with planned testing activities. The SQA test execution and reporting documentation will reside in the Rational Quality Manager (RQM) “EPIP” Project. In order to perform testing of this VistA intake product modification, the following tools will be leveraged: RQM, Reflections emulator, CPRS GUI v31 (1.0.30.75), and SnagIt.

## Test Readiness Review

Leidos will conduct a Test Readiness Review (TRR) at the conclusion of unit testing to verify the contents of the software to be tested, the test schedule, test environments, test participants, and associated logistics. Leidos will provide an agenda prior to the TRR and written minutes after completion of the TRR, in accordance with the Performance Work Statement (PWS).

## Testing Phases

Leidos will perform development and SQA testing activities in phases, and will provide all required testing documentation.

### Unit Testing

The developer will conduct unit testing of individual units of source code to determine if they are fit for use.

### Component Integration and Systems Testing (CI/ST)

Component integration and systems testing will be conducted by SQA to ensure that connectivity to the VistA application exists and is functioning normally. SQA will record Passed/Failed outcomes and capture displayed content to document the system testing effort.

### Functional Testing

Functional testing will be performed by SQA to test the code modifications. This testing will ensure that the software functionality is in alignment with the Government Furnished Information. SQA will record Passed/Failed outcomes and capture displayed content to document the functional testing effort.

### Regression Testing

Regression Testing will be performed by SQA to ensure that the remediated code does not introduce errors to existing functionality. The regression test framework will be kept up-to-date with manual test cases and test scripts defining the inputs and expected outcomes. SQA will record Passed/Failed outcomes and capture displayed content to document the regression testing effort.

### Section 508 Compliance Testing

508 Testing will be performed on VistA and CPRS code when new CPRS GUI changes are introduced by the developer. The VA-recommended Assistive Technology tool, JAWS, will be used to conduct the 508 testing. Test results and related documentation will be submitted to the VA Section 508 team in accordance with the VA 508 testing requirements. Defects found during testing will be assessed and remediated by the developer.

# Documentation Remediation

Leidos will review existing VA documentation for possible impact as a result of this remediation effort, and will make updates where applicable.

To determine the existing VA documentation that requires modification, Leidos will conduct a thorough review of the documents currently available from the VA Software Document Library (VDL) located at. Keyword searches using terms relevant to this remediation effort will be used to identify documents that might be impacted; those documents were will then be reviewed in their entirety for any needed revisions.

The following sections outline the VDL documents to be revised for this remediation.

## User Guides

The following User Guide will be updated in the VDL:

*Computerized Patient Record System (CPRS) User Manual: GUI Version* (Tentative)

## Installation Guides

The *National Patch Module Patch Description* document for this remediation will provide the procedure for installing KIDS packages migrated from the test environment to the VA   
Pre-Production environments. Therefore, no Installation Guides will be updated.

## Technical Manuals

The following Technical Manual will be updated in the VDL:

*Computerized Patient Record System (CPRS) Technical Manual: GUI Version*

## Operations Manuals

No Operations Manuals require revision as a result of this modification.

# Project Reporting

Leidos will provide interim progress updates during daily Scrum calls and weekly management calls with VA representatives.

# Project Schedule

Leidos will follow the Scrum Agile methodology for software development. It is anticipated that this patch will require five 2-week sprints.

# Deployment

Leidos will create a KIDS package containing the software changes necessary to fulfill the requirements for this remediation effort. A KIDS package, along with all related documentation, will be delivered to the Contracting Office Representative (COR) for acceptance. If accepted, the KIDS package can then be released for national VA consumption; otherwise, Leidos will correct any defects found and repeat the necessary remediation activities.

# Sustainment Requirements

Leidos will provide maintenance support for 60 days to the VA to support the final Class 1 product after it is nationally released.

# Maintenance and Knowledge Transfer

To facilitate continuous process improvement, Leidos will deliver *Sprint Review and Retrospective* slides and a *Lessons Learned* *Report* to VA upon completion of the final sprint.

1. XINDEX Listing for MUMPS Code Changes

The XINDEX tool is the standard tool used by the VA to analyze MUMPS source code. Following is a listing of the results of the XINDEX analysis of the affected routines.

**NSR20150103**

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:18:59

Routines: 1 Faux Routines: 0

ORQ12

--- CROSS REFERENCING ---

Press return to continue:

Compiled list of Errors and Warnings Dec 13, 2016@19:18:59 page 1

No errors or warnings to report

--- END ---

**NSR20160613**

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@14:56:21

Routines: 1 Faux Routines: 0

ORQOAUIA

--- CROSS REFERENCING ---

Press return to continue:

Compiled list of Errors and Warnings Dec 13, 2016@14:56:21 page 1

No errors or warnings to report

--- END ---

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:15:14

Routines: 1 Faux Routines: 0

ORQOAUIB

--- CROSS REFERENCING ---

Press return to continue

Compiled list of Errors and Warnings Dec 13, 2016@19:15:14 page 1

No errors or warnings to report

--- END ---

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:17:09

Routines: 1 Faux Routines: 0

ORQOAUIC

--- CROSS REFERENCING ---

Press return to continue:

Compiled list of Errors and Warnings Dec 13, 2016@19:17:09 page 1

No errors or warnings to report

--- END ---

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:20:15

Routines: 1 Faux Routines: 0

ORWDXM3

--- CROSS REFERENCING ---

Press return to continue:

Compiled list of Errors and Warnings Dec 13, 2016@19:20:15 page 1

No errors or warnings to report

--- END ---

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:21:19

All Routines? No => No

Routine: ORWPT

Routine:

1 routine

Select BUILD NAME:

Select INSTALL NAME:

Select PACKAGE NAME:

Print more than compiled errors and warnings? YES//N

Save parameters in ROUTINE file? NO//

Index all called routines? NO//

DEVICE: HOME (CRT) Right Margin: 80//

V. A. C R O S S R E F E R E N C E R 7.3

[2008 VA Standards & Conventions]

UCI: VISTA CPU: ROU Dec 13, 2016@19:21:19

Routines: 1 Faux Routines: 0

ORWPT

--- CROSS REFERENCING ---

Press return to continue:

Compiled list of Errors and Warnings Dec 13, 2016@19:21:19 page 1

No errors or warnings to report

--- END ---

1. Source Code Changes

This appendix displays the VistA code before and after the updates required for this code modification were implemented. The following routines were affected:

* **Modified routines:** ORQ12, ORWDXM3, ORWPT
* **New routines:** ORQOAUIA, ORQOAUIB, ORQOAUIC

**ORQ12**

**Before:**

ORQ12 ; slc/dcm - Get patient orders in context ;08/05/15 11:03  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*12,27,78,92,116,190,220,215,243,356,377\*\*;Dec 17, 1997;Build 299  
GET(IFN,NEWD,DETAIL,ACTOR) ; -- Setup TMP array  
 ; IFN=ifn of order  
 ; NEWD=3rd subscript in ^TMP("ORR",$J, node (ORLIST)  
 ; DETAIL=see description in ^ORQ1  
 ;  
 N X0,X3,X4,X6,TXT,STAT,START,DG,STOP,ENTERD  
 S ORLST=ORLST+1,^TMP("ORGOTIT",$J,IFN,+$G(ACTOR))=""  
 I '$G(DETAIL) S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"") Q  
 S X0=^OR(100,IFN,0),X3=$G(^(3)),X4=$G(^(4)),X6=$G(^(6))  
 S DG=$P(X0,U,11),DG=$P($G(^ORD(100.98,+DG,0)),U,3)  
 S STAT=$S($P(X3,U,3):$P(^ORD(100.01,$P(X3,U,3),0),U,1,2),1:"") ;.01^abbr  
 S ENTERD=$P(X0,U,7),START=$P(X0,U,8),STOP=$P(X0,U,9)  
 ; S FLAGREA=$P(X6,U,7)  
 S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"")\_U\_DG\_U\_ENTERD\_U\_START\_U\_STOP\_U\_STAT  
 D TEXT(.TXT,IFN\_";"\_$G(ACTOR)) M ^TMP("ORR",$J,NEWD,ORLST,"TX")=TXT  
 Q  
 ;  
TEXT(ORTX,ORIFN,WIDTH) ; -- Returns text of order ORIFN in ORTX(#)  
 N OR0,OR3,OR6,X,Y,FIRST,ORI,ORJ,DLG,ORX,ORACT,ORTA  
 K ORTX S:'$G(WIDTH) WIDTH=244  
 S ORACT=+$P(ORIFN,";",2),ORIFN=+ORIFN  
 I ORACT<1 S ORACT=+$P($G(^OR(100,ORIFN,3)),U,7) S:'ORACT ORACT=1  
 ;D:$O(^OR(100,ORIFN,1,0)) CNV^ORY92(ORIFN) ;convert text otf  
 S OR0=$G(^OR(100,ORIFN,0)),OR3=$G(^(3)),OR6=$G(^(6)),ORX=$G(^(8,ORACT,0))  
 S ORTX=1,ORTX(1)=""  
 I $P($G(OR0),U,11)'="",($P(^ORD(100.98,$P(OR0,U,11),0),U)="NON-VA MEDICATIONS") S X="Non-VA" D ADD  
 G:$G(ORIGVIEW)>1 T1  
 S:$P(OR0,U,14)=$O(^DIC(9.4,"C","OR",0)) ORTX(1)=">>" ;generic  
 S X=$$ACTION($P(ORX,U,2)) D:$L(X) ADD  
 I $P(ORX,U,2)="NW",$P(OR3,U,11),'$G(ORIGVIEW) D  ; Changed or Renewed  
 . I $P(OR3,U,11)=2 S X="Renew" D ADD Q  
 . N ORIG,ORIGTA S ORIG=+$P(OR3,U,5) Q:'ORIG  Q:$P(OR3,U,11)'=1  
 . S X="Change" D ADD S ORI=0  
 . I $G(IOST)'="P-OTHER" D  
 . .S ORIGTA=$$LASTXT(ORIG) ;D:$O(^OR(100,ORIG,1,0)) CNV^ORY92(ORIG)  
 . .F  S ORI=$O(^OR(100,ORIG,8,ORIGTA,.1,ORI)) Q:ORI'>0 S X=$G(^(ORI,0)) S:$E(X,1,3)=">> " X=$E(X,4,999) D ADD  
 . .S X=" to" D ADD  
T1 S ORTA=+$P(ORX,U,14),FIRST=+$O(^OR(100,ORIFN,8,ORTA,.1,0))  
 S ORI=0 F  S ORI=$O(^OR(100,ORIFN,8,ORTA,.1,ORI)) Q:ORI'>0 S X=$G(^(ORI,0)) S:(FIRST=ORI)&($E(X,1,3)=">> ") X=$E(X,4,999) D:$L(X) ADD  
 Q:$G(ORIGVIEW)>1 ;contents of global only  
 S DLG=$P(OR0,U,5) K Y I DLG,$P(DLG,";",2)["101.41",$D(^ORD(101.41,+DLG,9)) X ^(9) I $L($G(Y)) S X=Y D ADD ; additional text  
 ; I $P(OR3,U,11)=2 S X="(Renewal)" D ADD  
 I $P(ORX,U,4)=2 S X="\*UNSIGNED\*" D ADD  
 I $P(ORX,U,2)="DC"!("^1^13^"[(U\_$P(OR3,U,3)\_U)),$L(OR6) S X=" <"\_$S($L($P(OR6,U,5)):$P(OR6,U,5),$P(OR6,U,4):$P($G(^ORD(100.03,+$P(OR6,U,4),0)),U),1:"")\_">" D:$L(X)>3 ADD ; DC Reason  
 I $D(XQAID),$G(ORFLG)=12 S ORX=$G(^OR(100,ORIFN,8,ORACT,3)) I $P(ORX,U) S X=" Flagged "\_$$DATETIME($P(ORX,U,3))\_$S($P(ORX,U,4):" by "\_$$NAME($P(ORX,U,4)),1:"")\_": "\_$P(ORX,U,5) D ADD ; Flagged - show in FUP  
 Q  
 ;  
LASTXT(IFN) ; -- Returns action with latest text for order IFN  
 N I,Y S Y=1  
 S I=0 F  S I=$O(^OR(100,IFN,8,I)) Q:I'>0 S:$O(^(I,.1,0)) Y=I  
 Q Y  
 ;  
LAST(CODE) ; -- Return DA of last occurence of CODE action  
 N DA  
 I '$L($G(CODE)) S DA=$O(^OR(100,ORIFN,8,"A"),-1) ; last entry  
 E  S DA=$O(^OR(100,ORIFN,8,"C",CODE,"?"),-1) ; last CODE entry  
 Q DA  
 ;  
ACTION(X) ; -- Returns text of action X  
 N Y  
 S Y=$S(X="DC":"Discontinue",X="HD":"Hold",X="RL"&'$G(ORIGVIEW):"Release Hold of",X="FL":"Flag",X="UF":"Unflag",X="RN"&'$G(ORIGVIEW):"Renew",1:"")  
 Q Y  
 ;  
DATETIME(X) ; -- Returns date/time in format 00/00/00@00:00am  
 N Y,D,T,T1,Z  
 S D=$P(X,"."),T=$E($P(X,".",2)\_"0000",1,4),T1=$E(T,1,2),Z="AM"  
 S:T1>12 T1=T1-12,Z="PM"  
 S Y=$E(D,4,5)\_"/"\_$E(D,6,7)\_"/"\_(1700+$E(D,1,3))\_"@"\_T1\_":"\_$E(T,3,4)\_Z  
 Q Y  
 ;  
NAME(X) ; -- Returns name as Lname,F  
 N Y,Z S Z=$P($G(^VA(200,+X,0)),U) Q:Z="" ""  
 S Y=$P(Z,",")\_"," F I=$F(Z,","):1:$L(Z) I $E(Z,I)'=" " S Y=Y\_$E(Z,I) Q  
 S Y=$$LOWER^VALM1(Y) ; mixed case  
 Q Y  
 ;  
ADD ; -- Add text X to ORTX()  
 N I,Y S Y=$L(ORTX(ORTX)) S:Y Y=Y+1 ;allow for space  
 I $E(X)=" ",Y S ORTX=ORTX+1,ORTX(ORTX)="",Y=0,X=$E(X,2,999) ;new line  
 I Y+$L(X)'>WIDTH S ORTX(ORTX)=ORTX(ORTX)\_$S(Y:" ",1:"")\_X Q  
 F I=1:1:$L(X," ") S Z=$P(X," ",I) D:(Y+$L(Z))>WIDTH  S ORTX(ORTX)=$G(ORTX(ORTX))\_$S(Y:" ",1:"")\_Z,Y=$L(ORTX(ORTX)) S:Y Y=Y+1  
 . I $L(Z)>WIDTH F  S ORTX(ORTX)=$G(ORTX(ORTX))\_$S(Y:" ",1:"")\_$E(Z,1,WIDTH-Y),Z=$E(Z,WIDTH-Y+1,999) Q:$L(Z)'>WIDTH  S ORTX=ORTX+1,Y=0  
 . S ORTX=ORTX+1,Y=0  
 Q  
 ;  
EXPD ; -- loop through ^XTMP("ORAE" to get expired orders  
 K ^TMP("ORGOTIT",$J),^TMP("ORSORT",$J)  
 N TM,TO,IFN,X0,X3,X7,X8,USTS,NOW,ACTOR,X,ORREP  
 S NOW=+$E($$NOW^XLFDT,1,12),TO=0,SDATE=9999999-SDATE,EDATE=9999999-EDATE  
 F  S TO=$O(^XTMP("ORAE",PAT,TO)) Q:'TO  I $D(ORGRP(TO)) S TM=EDATE F  S TM=$O(^XTMP("ORAE",PAT,TO,TM)) Q:'TM!(TM>SDATE)!(+TM<EDATE) D  
 . S IFN=0 F  S IFN=$O(^XTMP("ORAE",PAT,TO,TM,IFN)) Q:'IFN  I ('$D(^TMP("ORGOTIT",$J,IFN))!MULT) D  
 .. ;\*356 Protect if x-ref dangles.  
 .. I '$D(^OR(100,IFN)) K ^XTMP("ORAE",PAT,TO,TM,IFN) Q  
 .. S USTS=$P(^OR(100,IFN,3),U,3)  
 .. Q:+$G(USTS)'=7 ;quit if order no longer expired  
 .. S ORREP=$P(^OR(100,IFN,3),U,6)  
 .. Q:+$G(ORREP)>0 ;quit if order has been replaced  
 .. S ^TMP("ORSORT",$J,9999999-TM,TO,IFN)=""  
 S TM=0 F  S TM=$O(^TMP("ORSORT",$J,TM)) Q:'TM  S TO=0 F  S TO=$O(^TMP("ORSORT",$J,TM,TO)) Q:'TO  D  
 .S IFN=0 F  S IFN=$O(^TMP("ORSORT",$J,TM,TO,IFN)) Q:'IFN  I $D(^OR(100,IFN,0)),$D(^(3)) S X0=^(0),X3=^(3) D  
 ..S ACTOR=+$P(X3,U,7) D LP1^ORQ11  
 ..;S ACTOR=0 F S ACTOR=$O(^OR(100,"ACT",PAT,9999999-$P(X0,U,7),TO,IFN,ACTOR)) Q:ACTOR<1 I '$D(^TMP("ORGOTIT",$J,IFN,ACTOR)),$D(^OR(100,IFN,8,ACTOR,0)),$P(^(0),U,15)'=13 S X8=^(0),X7=$G(^(7)) D LP1^ORQ11  
 S ^TMP("ORR",$J,ORLIST,"TOT")=$G(ORLST)  
 K ^TMP("ORSORT",$J),^TMP("ORGOTIT",$J)  
 Q  
GETEIE(IFN,NEWD,DETAIL,ACTOR) ; -- Setup TMP array  
 ; IFN=ifn of order  
 ; NEWD=3rd subscript in ^TMP("ORR",$J, node (ORLIST)  
 ; DETAIL=see description in ^ORQ1  
 ;  
 N X0,X3,X4,X6,TXT,STAT,START,DG,STOP,ENTERD,DCREAS  
 S X0=^OR(100,IFN,0),X3=$G(^(3)),X4=$G(^(4)),X6=$G(^(6))  
 S DG=$P(X0,U,11),DG=$P($G(^ORD(100.98,+DG,0)),U,3)  
 S STAT=$S($P(X3,U,3):$P(^ORD(100.01,$P(X3,U,3),0),U,1,2),1:"")  
 S ENTERD=$P(X0,U,7),START=$P(X0,U,8),STOP=$P(X0,U,9)  
 S DCREAS=$P($G(X6),U,4) Q:DCREAS'>0  
 I DCREAS'=$O(^ORD(100.03,"B","Entered in error","")) Q  
 S ORLST=ORLST+1,^TMP("ORGOTIT",$J,IFN,+$G(ACTOR))=""  
 I '$G(DETAIL) S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"") Q  
 S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"")\_U\_DG\_U\_ENTERD\_U\_START\_U\_STOP\_U\_STAT  
 D TEXT(.TXT,IFN) M ^TMP("ORR",$J,NEWD,ORLST,"TX")=TXT  
 Q

**After:**

ORQ12 ; slc/dcm - Get patient orders in context ;08/05/15 11:03  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*12,27,78,92,116,190,220,215,243,356,377\*\*;Dec 17, 1997;Build 299  
GET(IFN,NEWD,DETAIL,ACTOR) ; -- Setup TMP array  
 ; IFN=ifn of order  
 ; NEWD=3rd subscript in ^TMP("ORR",$J, node (ORLIST)  
 ; DETAIL=see description in ^ORQ1  
 ;  
 N X0,X3,X4,X6,TXT,STAT,START,DG,STOP,ENTERD  
 S ORLST=ORLST+1,^TMP("ORGOTIT",$J,IFN,+$G(ACTOR))=""  
 I '$G(DETAIL) S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"") Q  
 S X0=^OR(100,IFN,0),X3=$G(^(3)),X4=$G(^(4)),X6=$G(^(6))  
 S DG=$P(X0,U,11),DG=$P($G(^ORD(100.98,+DG,0)),U,3)  
 S STAT=$S($P(X3,U,3):$P(^ORD(100.01,$P(X3,U,3),0),U,1,2),1:"") ;.01^abbr  
 S ENTERD=$P(X0,U,7),START=$P(X0,U,8),STOP=$P(X0,U,9)  
 ; S FLAGREA=$P(X6,U,7)  
 S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"")\_U\_DG\_U\_ENTERD\_U\_START\_U\_STOP\_U\_STAT  
 D TEXT(.TXT,IFN\_";"\_$G(ACTOR)) M ^TMP("ORR",$J,NEWD,ORLST,"TX")=TXT  
 Q  
 ;  
TEXT(ORTX,ORIFN,WIDTH) ; -- Returns text of order ORIFN in ORTX(#)  
 N OR0,OR3,OR6,X,Y,FIRST,ORI,ORJ,DLG,ORX,ORACT,ORTA  
 K ORTX S:'$G(WIDTH) WIDTH=244  
 S ORACT=+$P(ORIFN,";",2),ORIFN=+ORIFN  
 I ORACT<1 S ORACT=+$P($G(^OR(100,ORIFN,3)),U,7) S:'ORACT ORACT=1  
 ;D:$O(^OR(100,ORIFN,1,0)) CNV^ORY92(ORIFN) ;convert text otf  
 S OR0=$G(^OR(100,ORIFN,0)),OR3=$G(^(3)),OR6=$G(^(6)),ORX=$G(^(8,ORACT,0))  
 S ORTX=1,ORTX(1)=""  
 I $P($G(OR0),U,11)'="",($P(^ORD(100.98,$P(OR0,U,11),0),U)="NON-VA MEDICATIONS") S X="Non-VA" D ADD  
 G:$G(ORIGVIEW)>1 T1  
 S:$P(OR0,U,14)=$O(^DIC(9.4,"C","OR",0)) ORTX(1)=">>" ;generic  
 S X=$$ACTION($P(ORX,U,2)) D:$L(X) ADD  
 I $P(ORX,U,2)="NW",$P(OR3,U,11),'$G(ORIGVIEW) D  ; Changed or Renewed  
 . I $P(OR3,U,11)=2 S X="Renew" D ADD Q  
 . N ORIG,ORIGTA S ORIG=+$P(OR3,U,5) Q:'ORIG  Q:$P(OR3,U,11)'=1  
 . S X="Change" D ADD S ORI=0  
 . I $G(IOST)'="P-OTHER" D  
 . .S ORIGTA=$$LASTXT(ORIG) ;D:$O(^OR(100,ORIG,1,0)) CNV^ORY92(ORIG)  
 . .F  S ORI=$O(^OR(100,ORIG,8,ORIGTA,.1,ORI)) Q:ORI'>0 S X=$G(^(ORI,0)) S:$E(X,1,3)=">> " X=$E(X,4,999) D ADD  
 . .S X=" to" D ADD  
T1 S ORTA=+$P(ORX,U,14),FIRST=+$O(^OR(100,ORIFN,8,ORTA,.1,0))  
 S ORI=0 F  S ORI=$O(^OR(100,ORIFN,8,ORTA,.1,ORI)) Q:ORI'>0 S X=$G(^(ORI,0)) S:(FIRST=ORI)&($E(X,1,3)=">> ") X=$E(X,4,999) D:$L(X) ADD  
 Q:$G(ORIGVIEW)>1 ;contents of global only  
 S DLG=$P(OR0,U,5) K Y I DLG,$P(DLG,";",2)["101.41",$D(^ORD(101.41,+DLG,9)) X ^(9) I $L($G(Y)) S X=Y D ADD ; additional text  
 ; I $P(OR3,U,11)=2 S X="(Renewal)" D ADD  
 I $P(ORX,U,4)=2 S X="\*UNSIGNED\*" D ADD  
 S ORXZ=$D(^OR(100,ORIFN,8,ORACT,5,0)) I ORXZ S X=$G(^OR(100,ORIFN,8,ORACT,5,1,0)) D:$L(X) ADD K ORXZ ;adds Ward Comments  
 I $P(ORX,U,2)="DC"!("^1^13^"[(U\_$P(OR3,U,3)\_U)),$L(OR6) S X=" <"\_$S($L($P(OR6,U,5)):$P(OR6,U,5),$P(OR6,U,4):$P($G(^ORD(100.03,+$P(OR6,U,4),0)),U),1:"")\_">" D:$L(X)>3 ADD ; DC Reason  
 I +$G(^OR(100,ORIFN,8,ORACT,3)),$L($P(^OR(100,ORIFN,8,ORACT,3),U,5)) S X="\*Flagged - "\_$P(^OR(100,ORIFN,8,ORACT,3),U,5)\_" - " D ADD   
 I $D(XQAID),$G(ORFLG)=12 S ORX=$G(^OR(100,ORIFN,8,ORACT,3)) I $P(ORX,U) S X=" Flagged "\_$$DATETIME($P(ORX,U,3))\_$S($P(ORX,U,4):" by "\_$$NAME($P(ORX,U,4)),1:"")\_": "\_$P(ORX,U,5) D ADD ; Flagged - show in FUP  
 Q  
 ;  
LASTXT(IFN) ; -- Returns action with latest text for order IFN  
 N I,Y S Y=1  
 S I=0 F  S I=$O(^OR(100,IFN,8,I)) Q:I'>0 S:$O(^(I,.1,0)) Y=I  
 Q Y  
 ;  
LAST(CODE) ; -- Return DA of last occurence of CODE action  
 N DA  
 I '$L($G(CODE)) S DA=$O(^OR(100,ORIFN,8,"A"),-1) ; last entry  
 E  S DA=$O(^OR(100,ORIFN,8,"C",CODE,"?"),-1) ; last CODE entry  
 Q DA  
 ;  
ACTION(X) ; -- Returns text of action X  
 N Y  
 S Y=$S(X="DC":"Discontinue",X="HD":"Hold",X="RL"&'$G(ORIGVIEW):"Release Hold of",X="FL":"Flag",X="UF":"Unflag",X="RN"&'$G(ORIGVIEW):"Renew",1:"")  
 Q Y  
 ;  
DATETIME(X) ; -- Returns date/time in format 00/00/00@00:00am  
 N Y,D,T,T1,Z  
 S D=$P(X,"."),T=$E($P(X,".",2)\_"0000",1,4),T1=$E(T,1,2),Z="AM"  
 S:T1>12 T1=T1-12,Z="PM"  
 S Y=$E(D,4,5)\_"/"\_$E(D,6,7)\_"/"\_(1700+$E(D,1,3))\_"@"\_T1\_":"\_$E(T,3,4)\_Z  
 Q Y  
 ;  
NAME(X) ; -- Returns name as Lname,F  
 N Y,Z S Z=$P($G(^VA(200,+X,0)),U) Q:Z="" ""  
 S Y=$P(Z,",")\_"," F I=$F(Z,","):1:$L(Z) I $E(Z,I)'=" " S Y=Y\_$E(Z,I) Q  
 S Y=$$LOWER^VALM1(Y) ; mixed case  
 Q Y  
 ;  
ADD ; -- Add text X to ORTX()  
 N I,Y S Y=$L(ORTX(ORTX)) S:Y Y=Y+1 ;allow for space  
 I $E(X)=" ",Y S ORTX=ORTX+1,ORTX(ORTX)="",Y=0,X=$E(X,2,999) ;new line  
 I Y+$L(X)'>WIDTH S ORTX(ORTX)=ORTX(ORTX)\_$S(Y:" ",1:"")\_X Q  
 F I=1:1:$L(X," ") S Z=$P(X," ",I) D:(Y+$L(Z))>WIDTH  S ORTX(ORTX)=$G(ORTX(ORTX))\_$S(Y:" ",1:"")\_Z,Y=$L(ORTX(ORTX)) S:Y Y=Y+1  
 . I $L(Z)>WIDTH F  S ORTX(ORTX)=$G(ORTX(ORTX))\_$S(Y:" ",1:"")\_$E(Z,1,WIDTH-Y),Z=$E(Z,WIDTH-Y+1,999) Q:$L(Z)'>WIDTH  S ORTX=ORTX+1,Y=0  
 . S ORTX=ORTX+1,Y=0  
 Q  
 ;  
EXPD ; -- loop through ^XTMP("ORAE" to get expired orders  
 K ^TMP("ORGOTIT",$J),^TMP("ORSORT",$J)  
 N TM,TO,IFN,X0,X3,X7,X8,USTS,NOW,ACTOR,X,ORREP  
 S NOW=+$E($$NOW^XLFDT,1,12),TO=0,SDATE=9999999-SDATE,EDATE=9999999-EDATE  
 F  S TO=$O(^XTMP("ORAE",PAT,TO)) Q:'TO  I $D(ORGRP(TO)) S TM=EDATE F  S TM=$O(^XTMP("ORAE",PAT,TO,TM)) Q:'TM!(TM>SDATE)!(+TM<EDATE) D  
 . S IFN=0 F  S IFN=$O(^XTMP("ORAE",PAT,TO,TM,IFN)) Q:'IFN  I ('$D(^TMP("ORGOTIT",$J,IFN))!MULT) D  
 .. ;\*356 Protect if x-ref dangles.  
 .. I '$D(^OR(100,IFN)) K ^XTMP("ORAE",PAT,TO,TM,IFN) Q  
 .. S USTS=$P(^OR(100,IFN,3),U,3)  
 .. Q:+$G(USTS)'=7 ;quit if order no longer expired  
 .. S ORREP=$P(^OR(100,IFN,3),U,6)  
 .. Q:+$G(ORREP)>0 ;quit if order has been replaced  
 .. S ^TMP("ORSORT",$J,9999999-TM,TO,IFN)=""  
 S TM=0 F  S TM=$O(^TMP("ORSORT",$J,TM)) Q:'TM  S TO=0 F  S TO=$O(^TMP("ORSORT",$J,TM,TO)) Q:'TO  D  
 .S IFN=0 F  S IFN=$O(^TMP("ORSORT",$J,TM,TO,IFN)) Q:'IFN  I $D(^OR(100,IFN,0)),$D(^(3)) S X0=^(0),X3=^(3) D  
 ..S ACTOR=+$P(X3,U,7) D LP1^ORQ11  
 ..;S ACTOR=0 F S ACTOR=$O(^OR(100,"ACT",PAT,9999999-$P(X0,U,7),TO,IFN,ACTOR)) Q:ACTOR<1 I '$D(^TMP("ORGOTIT",$J,IFN,ACTOR)),$D(^OR(100,IFN,8,ACTOR,0)),$P(^(0),U,15)'=13 S X8=^(0),X7=$G(^(7)) D LP1^ORQ11  
 S ^TMP("ORR",$J,ORLIST,"TOT")=$G(ORLST)  
 K ^TMP("ORSORT",$J),^TMP("ORGOTIT",$J)  
 Q  
GETEIE(IFN,NEWD,DETAIL,ACTOR) ; -- Setup TMP array  
 ; IFN=ifn of order  
 ; NEWD=3rd subscript in ^TMP("ORR",$J, node (ORLIST)  
 ; DETAIL=see description in ^ORQ1  
 ;  
 N X0,X3,X4,X6,TXT,STAT,START,DG,STOP,ENTERD,DCREAS  
 S X0=^OR(100,IFN,0),X3=$G(^(3)),X4=$G(^(4)),X6=$G(^(6))  
 S DG=$P(X0,U,11),DG=$P($G(^ORD(100.98,+DG,0)),U,3)  
 S STAT=$S($P(X3,U,3):$P(^ORD(100.01,$P(X3,U,3),0),U,1,2),1:"")  
 S ENTERD=$P(X0,U,7),START=$P(X0,U,8),STOP=$P(X0,U,9)  
 S DCREAS=$P($G(X6),U,4) Q:DCREAS'>0  
 I DCREAS'=$O(^ORD(100.03,"B","Entered in error","")) Q  
 S ORLST=ORLST+1,^TMP("ORGOTIT",$J,IFN,+$G(ACTOR))=""  
 I '$G(DETAIL) S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"") Q  
 S ^TMP("ORR",$J,NEWD,ORLST)=IFN\_$S($G(ACTOR):";"\_ACTOR,1:"")\_U\_DG\_U\_ENTERD\_U\_START\_U\_STOP\_U\_STAT  
 D TEXT(.TXT,IFN) M ^TMP("ORR",$J,NEWD,ORLST,"TX")=TXT  
 Q

**ORWDXM3**

**Before:**

ORWDXM3 ; SLC/KCM/JLI - Quick Orders ;08/13/15 20:49  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*10,85,131,132,141,185,187,190,195,215,243,303,296,280,350,377\*\*;Dec 17, 1997;Build 299  
 ;  
VALCOUNT(NAME,ORDIALOG) ;  
 N COUNT,IEN,NUM  
 S NUM=0,COUNT=0  
 S IEN=$P($G(ORDIALOG("B",NAME)),U,2) Q:IEN'>0  
 F  S NUM=$O(ORDIALOG(IEN,NUM)) Q:+NUM'>0 S COUNT=COUNT+1  
 Q COUNT  
 ;  
ISMISSFL(ORDIALOG,IVTYPE) ;  
 N ADDCNT,RESULT,SOLCNT,STRCNT  
 S RESULT=0  
 S ADDCNT=$$VALCOUNT("ADDITIVE",.ORDIALOG)  
 S STRCNT=$$VALCOUNT("STRENGTH",.ORDIALOG)  
 S SOLCNT=$$VALCOUNT("SOLUTION",.ORDIALOG)  
 I IVTYPE'="I",ADDCNT'=STRCNT S RESULT=1  
 I IVTYPE="I" D  
 .I ADDCNT=0,SOLCNT>0 Q  
 .I ADDCNT=0 S RESULT=1 Q  
 .I ADDCNT'=STRCNT S RESULT=1 Q  
 Q RESULT  
 ;  
IVADFCHK(ORDIALOG) ;  
 ; This line tag checks to see if there are the same number of values  
 ;for ADDITIVE and Additive Frequency. This also checks to see if  
 ;the value assigned to ADDFREQ is one of the three possible values  
 ;All Bags, 1 bag/day, See Comments. If See Comments it also checks  
 ;for text in the comment section.  
 N ADDCNT,ADDFCNT,ADDFREQ,COMMENT,FREQ,INST,RESULT  
 S ADDCNT=$$VALCOUNT("ADDITIVE",.ORDIALOG)  
 S ADDFCNT=$$VALCOUNT("ADDITIVE FREQUENCY",.ORDIALOG)  
 I ADDCNT'=ADDFCNT Q 0  
 S ADDFREQ=$O(^ORD(101.41,"AB","OR GTX ADDITIVE FREQUENCY",""))  
 S COMMENT=$O(^ORD(101.41,"AB","OR GTX WORD PROCESSING 1",""))  
 I +$G(ADDFREQ)'>0 Q O  
 S INST=0,RESULT=1  
 F  S INST=$O(ORDIALOG(ADDFREQ,INST)) Q:INST'>0!(RESULT=0) D  
 .S FREQ=$$ADDFRQCV^ORMBLDP1($G(ORDIALOG(ADDFREQ,INST)),"O")  
 .I FREQ="A"!(FREQ=1) Q  
 .I FREQ="" S RESULT=0 Q  
 .I FREQ="S",'$L($G(ORDIALOG(COMMENT,1))) S RESULT=0 Q  
 Q RESULT  
 ;  
KEYVAR(DLG) ; Parse entry action for key variables & return in string  
 ; RV=CollTp^Samp^Spec^CollDt^Urg^Sched^NoComm^NoDiag^NoProv^NoRsn  
 N XCODE,RV,POS,Z  
 S XCODE=$G(^ORD(101.41,DLG,3)),RV=""  
 I '$L(XCODE) Q ""  
 S POS=$F(XCODE,"LRFZX=") I POS S $P(RV,U,1)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSAMP=") I POS S $P(RV,U,2)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSPEC=") I POS S $P(RV,U,3)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFDATE=") I POS S $P(RV,U,4)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFURG=") I POS S $P(RV,U,5)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSCH=") I POS S $P(RV,U,6)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"PSJNOPC=") I POS S $P(RV,U,7)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCNOPD=") I POS S $P(RV,U,8)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCNOAT=") I POS S $P(RV,U,9)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCREAF=") I POS S $P(RV,U,10)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"ORFORGET=") I POS D  
 . ; need to change this so that it is executed in SETKEYV so  
 . ; that it is executed each time menu is revisited  
 . N ORFORGET S ORFORGET=$$VALUE(XCODE,POS)  
 . I ORFORGET K ^TMP("ORECALL",$J,+ORFORGET)  
 . E  K ^TMP("ORECALL",$J)  
 Q RV  
VALUE(STR,BEG) ; Return value of "var=" (copied from ORCONVRT)  
 N X,Y,I S X=$E(STR,BEG,999),Y=""  
 S:$E(X)="""" X=$E(X,2,999) ; strip leading "  
 F I=1:1:$L(X) S Z=$E(X,I) Q:(Z=",")!(Z=" ")!(Z="""") S Y=Y\_Z  
 Q $TR(Y,U,"")  
 ;  
SETKEYV(X) ; Set the key variables based on contents of X  
 I $L($P(X,U,1)) S LRFZX=$P(X,U,1)  
 I $L($P(X,U,2)) S LRFSAMP=$P(X,U,2)  
 I $L($P(X,U,3)) S LRFSPEC=$P(X,U,3)  
 I $L($P(X,U,4)) S LRFDATE=$P(X,U,4)  
 I $L($P(X,U,5)) S LRFURG=$P(X,U,5)  
 I $L($P(X,U,6)) S LRFSCH=$P(X,U,6)  
 I $L($P(X,U,7)) S PSJNOPC=$P(X,U,7)  
 I $L($P(X,U,8)) S GMRCNOPD=$P(X,U,8)  
 I $L($P(X,U,9)) S GMRCNOAT=$P(X,U,9)  
 I $L($P(X,U,10)) S GMRCREAF=$P(X,U,10)  
 Q  
DLGINFO(IEN,MODE) ; return information about a dialog  
 ; IEN=DlgIEN or ORIFN, MODE=0:Dlg,1:Copy,2:Change  
 ; RESULT=DlgIEN^DlgType^FormID^DGrp  
 ; If MODE="1;T",don't check "PS MEDS" for transfer order  
 ; PSMDGP=1: Unit/Dose Group  
 ; PSMDGP=2: OutPatient Group  
 N X0,DLGIEN,TYP,FID,DGRP,PSMDGP,ISXF  
 S PSMDGP=0,ISXF=""  
 S ISXF=$P(MODE,";",2)  
 S MODE=+MODE  
 S DLGIEN=IEN I MODE,(ISXF'="T") D  
 . S DLGIEN=+$P($G(^OR(100,+IEN,0)),U,5)  
 . I $P(^ORD(101.41,DLGIEN,0),U)="PS MEDS" D  
 . . N PTCAT S PTCAT=$P($G(^OR(100,+IEN,0)),U,12)  
 . . I PTCAT="I" S DLGIEN=$O(^ORD(101.41,"B","PSJ OR PAT OE",0)),PSMDGP=1  
 . . I PTCAT="O" S DLGIEN=$O(^ORD(101.41,"B","PSO OERR",0)),PSMDGP=2  
 I MODE,(ISXF="T") S DLGIEN=+$P($G(^OR(100,+IEN,0)),U,5)  
 S X0=$G(^ORD(101.41,DLGIEN,0)),TYP=$P(X0,U,4),DGRP=$P(X0,U,5)  
 I MODE S DGRP=+$P($G(^OR(100,+IEN,0)),U,11)  
 ;JD NEW START 11/13/02  
 I DLGIEN=$O(^ORD(101.41,"B","PSJ OR PAT OE",0)) S PSMDGP=1  
 I DLGIEN=$O(^ORD(101.41,"B","PSO OERR",0)) S PSMDGP=2  
 ;JD NEW END 11/13/02  
 ; for copy or change, if the base dialog has changed, use it's info  
 I MODE,$G(ORDIALOG),(+DLGIEN'=+ORDIALOG),(PSMDGP=0) D  
 . S DLGIEN=+ORDIALOG,DGRP=$P(^ORD(101.41,+ORDIALOG,0),U,5)  
 D FORMID^ORWDXM(.FID,DLGIEN)  
 Q DLGIEN\_U\_TYP\_U\_FID\_U\_DGRP  
 ;  
CHKDSBL(LST,ID,MODE) ; return message if dialog disabled  
 ; ID=DlgIEN or ORIFN, MODE=0:Dialog,1:Copy,2:Change  
 ; LST=QL\_REJECT + disabled message or unchanged  
 S DLGIEN=+ID I MODE S DLGIEN=+$P($G(^OR(100,+ID,0)),U,5)  
 S X0=$G(^ORD(101.41,DLGIEN,0)),X=$P(X0,U,3)  
 I '$L(X),($P(X0,U,4)="Q") D  ; check default dialog  
 . S DLGIEN=+$$DEFDLG^ORWDXQ($P(X0,U,5))  
 . S X=$P($G(^ORD(101.41,DLGIEN,0)),U,3)  
 I $L(X) D  
 . I MODE D GETTXT^ORWORR(.LST,ID) S LST(.6)="",LST(.7)="Cannot "\_$S(MODE=1:"Copy",1:"Change")\_" -"  
 . S LST(0)="8^0",LST(.5)="Dialog Disabled: "\_X  
 Q  
CHKVACT(LST,ID,MODE,ORNP) ; return message if action not valid  
 ; ID=DlgIEN or ORIFN, MODE=0:Dialog,1:Copy,2:Change  
 ; LST=QL\_REJECT + invalid action message or unchanged  
 Q:'MODE  ; not an action on an order  
 N X,ACT S ACT=$S(MODE=1:"RW",MODE=2:"XX",1:"")  
 D VALID^ORWDXA(.X,ID,ACT,ORNP)  
 I $L(X) D GETTXT^ORWORR(.LST,ID) D  
 . S LST(0)="8^0",LST(.5)=X,LST(.6)="",LST(.7)="Cannot "\_$S(MODE=1:"Copy",1:"Change")\_" -"  
 Q  
CHKCOPY(LST,ID,FLDS) ; return message if can't copy this order  
 ; ID=ORIFN;ACT FLDS=EventType in 7th piece  
 ; LST=QL\_REJECT + cannot copy message or unchanged  
 I "^A^D^T^"'[(U\_$E($P(FLDS,U,7))\_U) Q             ; not event delayed  
 N PKG S PKG=$P($G(^OR(100,+ID,0)),U,14)  
 S PKG=$$NMSP^ORCD(PKG) I PKG="OR"!(PKG="PS") Q    ; xfer meds, generics  
 N ORWCAT S ORWCAT=$P($G(^OR(100,+ID,0)),U,12)  
 I ORWCAT="I",("^A^T^"[(U\_$E($P(FLDS,U,7))\_U)) Q   ; admit, xfer inpt  
 I ORWCAT="O",$E($P(FLDS,U,7))="D" Q               ; discharge outpt  
 D GETTXT^ORWORR(.LST,ID)  
 I ORWCAT="I" S LST(.5)="inpatient order to outpatient -"  
 I ORWCAT="O" S LST(.5)="outpatient order to inpatient -"  
 S:$D(LST(.5)) LST(.5)="Cannot copy the following "\_LST(.5)  
 S LST(0)="8^0",LST(.7)=""  
 Q  
BLD4CHG(LST,ID,FLDS) ; build responses for an edit  
 ; ID=ORIFN;ACT FLDS=unused right now  
 ; LST(0)=Qlvl^RespID(XOrderID)^DlgIEN^DlgType^FormID^DGrp  
 N OIDX,OI,CNT  
 S (OI,OIDX,CNT)=0  
 S:$D(^OR(100,+ID,4.5,"ID","ORDERABLE")) OIDX=$O(^OR(100,+ID,4.5,"ID","ORDERABLE",0))  
 I $D(^OR(100,+ID,4.5,OIDX)) D  
 . F  S CNT=$O(^OR(100,+ID,4.5,OIDX,CNT)) Q:'CNT  D  
 . . S OI=^(CNT) D VALDOI  
 I +LST(0)=8 S LST(.5)="You can not change this order." Q  
 S LST(0)="0^X"\_ID\_U\_$$DLGINFO(+ID,2)  
 S $P(LST(0),U,4)="X"  
 Q  
GETIVTYP() ;  
 N RESULT,TYPEIEN  
 S RESULT=""  
 S TYPEIEN=$O(^ORD(101.41,"B","OR GTX IV TYPE","")) I TYPEIEN'>0 Q RESULT  
 S RESULT=$G(ORDIALOG(TYPEIEN,1))  
 Q RESULT  
 ;  
ISTUBEQO(IFN) ;  
 N DG,DIAL  
 S DG=+$P($G(^ORD(101.41,IFN,0)),U,5)  
 S DIAL=$P($G(^ORD(100.98,DG,0)),U,4)  
 I +$G(DIAL)=0 Q 0  
 I $P($G(^ORD(101.41,DIAL,0)),U)="FHW8" Q 1  
 Q 0  
 ;  
VALDOI ; Validate the Orderable Items  
 N ORQUIT,ORPS  
 I $G(^ORD(101.43,OI,.1)),^(.1)'>$$NOW^XLFDT D  
 . S ORQUIT=1  
 . S LST(0)="8^0"  
 I $D(ORQUIT) Q:ORQUIT  
 S ORPS=$G(^ORD(101.43,+OI,"PS"))  
 I $P(ORPS,U,1,4)="0^0^0^0",($P(ORPS,U,7)=0) S LST(0)="8^0"  
 Q  
VERDUR(ORDIALOG) ;  
 ;check for duration value if a THEN conjunation is used  
 N CONJ,CONVALUE,DUR,I,SUCC  
 S SUCC=1  
 S CONJ=$$PTR^ORCDPS1("AND/THEN")  
 S DUR=$$PTR^ORCDPS1("DURATION")  
 S I=0 F  S I=$O(ORDIALOG(CONJ,I)) Q:I'>0!(SUCC=0) D  
 . I $$UP^XLFSTR($E($G(ORDIALOG(CONJ,I)),1))="T" D  
 . . I '$L($G(ORDIALOG(DUR,I))) S SUCC=0  
 Q SUCC  
 ;  
VERORD(OIEN) ;  
 N IFN,INFUSE,INFUID,ODG,ODP,ASSIV,SUCC,TYPE  
 S SUCC=0,IFN=ORDIALOG  
 S ODP=+$P($G(^ORD(101.41,+IFN,0)),U,7),ODG=+$P($G(^(0)),U,5)  
 S ODP=$$GET1^DIQ(9.4,+ODP\_",",1),ODG=$P($G(^ORD(100.98,ODG,0)),U,3)  
 I ODP'["PS" Q 1  
 I ODP="PSH" Q 1  
 ;check infusion rate for IV QO  
 I ODG="IV RX"!(ODG="TPN") D  Q SUCC  
 .S TYPE=$$GETIVTYP  
 .I TYPE="" Q  
 .S PASSIV=$$IVRTECHK  
 .I PASSIV=0 Q  
 .S INFUID=$O(^ORD(101.41,"B","OR GTX INFUSION RATE",0))  
 .S INFUSE=$G(ORDIALOG(INFUID,1))  
 .S SUCC=$$VALINF(TYPE,INFUSE)  
 .I SUCC=0 Q  
 .I TYPE="C" S SUCC=$$IVADFCHK(.ORDIALOG)  
 I (ODP="PSJ")!(ODP="PSO"),ODG'="IV RX",ODG'="TPN" S SUCC=$$VERDUR(.ORDIALOG)  
 Q SUCC  
 ;  
VALINF(TYPE,INFUSE) ;  
 N SUCC  
 S SUCC=0  
 I TYPE="I" D  Q SUCC  
 .I INFUSE="" S SUCC=1 Q  
 .I $TR(INFUSE,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")["INFUSE OVER" S SUCC=1 Q  
 .I $L(INFUSE)>4 Q  
 Q 1  
 ;  
VALQO(IFN) ;Check to see if it's a good QO med  
 ;If it's an IV QO: check if infusion rate entered  
 ;If it's an UD QO: check if dosage entered  
 ;regular order treated as good QO  
 ;  
 I IFN[";",($$UPCTCHK^ORWDXA(+IFN)) Q 0  
 I $P($G(^ORD(101.41,IFN,0)),U,4)'="Q" Q 1  
 N ODP,ODG,INFUID,INFUSE,DSAGEID,SUCC,PASSIV,TYPE  
 S SUCC=0  
 S ODP=+$P($G(^ORD(101.41,IFN,0)),U,7),ODG=+$P($G(^(0)),U,5)  
 S ODP=$$GET1^DIQ(9.4,+ODP\_",",1),ODG=$P($G(^ORD(100.98,ODG,0)),U,3)  
 I ODP'["PS" Q 1  
 I ODP="PSH" Q 1  
 ;check infusion rate for IV QO  
 I ODG="IV RX"!(ODG="TPN") D  
 . S INFUID=$O(^ORD(101.41,"B","OR GTX INFUSION RATE",0))  
 . S TYPE=$$GETIVTYP  
 . I TYPE="" Q  
 . I $D(ORDIALOG(INFUID,1)) D  
 . . I TYPE="I" D  Q  
 . . . S INFUSE=$G(ORDIALOG(INFUID,1))  
 . . . I INFUSE="" Q  
 . . . I INFUSE["INFUSE OVER" S SUCC=1 Q  
 . . . I $L(INFUSE)>4 Q  
 . . . I +INFUSE>0 S INFUSE="INFUSE OVER "\_INFUSE\_" Minutes"  
 . . . S ORDIALOG(INFUID,1)=INFUSE,SUCC=1  
 . . S SUCC=1  
 . ; additive frequency check/infusion rate checks for continuous orders  
 . I TYPE="C" D  I SUCC=0 Q  
 . . I $D(ORDIALOG(INFUID,1)) S SUCC=1 I SUCC=0 Q  
 . . S SUCC=$$IVADFCHK(.ORDIALOG)  
 . I SUCC=0 Q  
 . I '$D(ORDIALOG(INFUID,1)),TYPE="I" S SUCC=1  
 . S PASSIV=$$IVRTECHK  
 . I SUCC=0 Q  
 . I PASSIV=0 S SUCC=0  
 . I SUCC=1,$$ISMISSFL(.ORDIALOG,TYPE)=1 S SUCC=0  
 ;check dosage for UD QO  
 I (ODP="PSJ")!(ODP="PSO"),ODG'="IV RX",ODG'="TPN" D  
 . S DSAGEID=$O(^ORD(101.41,"B","OR GTX INSTRUCTIONS",0))  
 . I $D(ORDIALOG(DSAGEID,1)) S SUCC=1  
 . I SUCC=0 Q  
 . ;  
 . S SUCC=$$VERDUR(.ORDIALOG)  
 ;  
 I SUCC=1,$P($G(^ORD(101.41,IFN,5)),U,8) D  
 .N COMMID,WPCNT  
 .S COMMID=$O(^ORD(101.41,"B","OR GTX WORD PROCESSING 1",0))  
 .S COMMID=$O(^ORD(101.41,IFN,6,"D",COMMID,0))  
 .I COMMID S WPCNT=0 F  S WPCNT=$O(^ORD(101.41,IFN,6,COMMID,2,WPCNT)) Q:'WPCNT!('SUCC) D  
 ..I ^ORD(101.41,IFN,6,COMMID,2,WPCNT,0)["^" S SUCC=0  
 Q SUCC  
 ;  
IVRTECHK() ;  
 N RTIEN,RTVALUE,RESULT  
 N CNT,NUM,ORDERIDS,OIIEN,OTYPE,ROUTE  
 S CNT=0,RESULT=0  
 S RTIEN=+$P($G(ORDIALOG("B","ROUTE")),U,2) I RTIEN'>0 Q RESULT  
 S RTVALUE=+$G(ORDIALOG(RTIEN,1)) I RTVALUE'>0 Q RESULT  
 F OTYPE="SOLUTION","ADDITIVE" D  
 .S OIIEN=+$P($G(ORDIALOG("B",OTYPE)),U,2) I OIIEN>0 D  
 ..S NUM=0 F  S NUM=$O(ORDIALOG(OIIEN,NUM)) Q:NUM'>0 I +$G(ORDIALOG(OIIEN,NUM))>0 D  
 ...S CNT=CNT+1,ORDERIDS(CNT)=ORDIALOG(OIIEN,NUM)  
 I $D(ORDERIDS)=0 Q  
 S ROUTE=$$IVQOVAL^ORWDPS33(.ORDERIDS,RTVALUE)  
 I ROUTE="" S ORDIALOG(RTIEN,1)=ROUTE  
 I ROUTE'="" S RESULT=1  
 ;K ^TMP($J,"ORWDXM3 IVRTECHK")  
 ;D ALL^PSS51P2(RTVALUE,,,,"ORWDXM3 IVRTECHK")  
 ;I +^TMP($J,"ORWDXM3 IVRTECHK",RTVALUE,6)'=1 S ORDIALOG(RTIEN,1)="",RESULT=0  
 ;K ^TMP($J,"ORWDXM3 IVRTECHK")  
 Q RESULT  
 ;  
ISUDQO(ORY,DLGID) ;True: is unit dose quick order  
 S ORY=0  
 Q:'$D(^ORD(101.41,DLGID,0))  
 N CLODGRP,CLIVDGRP,UDGRP1,UDGRP2,DLGTYP,DLGGRP  
 S UDGRP1=$O(^ORD(100.98,"B","UD RX",0))  
 S UDGRP2=$O(^ORD(100.98,"B","I RX",0))  
 S CLODGRP=$O(^ORD(100.98,"B","CLINIC MEDICATIONS",""))  
 S CLIVDGRP=$O(^ORD(100.98,"B","CLINIC INFUSIONS",""))  
 S DLGTYP=$P($G(^ORD(101.41,DLGID,0)),U,4)

 S DLGGRP=$P($G(^ORD(101.41,DLGID,0)),U,5)  
 I (DLGTYP="Q"),((DLGGRP=UDGRP1)!(DLGGRP=UDGRP2)!(DLGGRP=CLODGRP)!(DLGGRP=CLIVDGRP)) S ORY=1  
 Q

**After:**

ORWDXM3 ; SLC/KCM/JLI - Quick Orders ;08/13/15 20:49  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*10,85,131,132,141,185,187,190,195,215,243,303,296,280,350,377\*\*;Dec 17, 1997;Build 299  
 ;  
VALCOUNT(NAME,ORDIALOG) ;  
 N COUNT,IEN,NUM  
 S NUM=0,COUNT=0  
 S IEN=$P($G(ORDIALOG("B",NAME)),U,2) Q:IEN'>0  
 F  S NUM=$O(ORDIALOG(IEN,NUM)) Q:+NUM'>0 S COUNT=COUNT+1  
 Q COUNT  
 ;  
ISMISSFL(ORDIALOG,IVTYPE) ;  
 N ADDCNT,RESULT,SOLCNT,STRCNT  
 S RESULT=0  
 S ADDCNT=$$VALCOUNT("ADDITIVE",.ORDIALOG)  
 S STRCNT=$$VALCOUNT("STRENGTH",.ORDIALOG)  
 S SOLCNT=$$VALCOUNT("SOLUTION",.ORDIALOG)  
 I IVTYPE'="I",ADDCNT'=STRCNT S RESULT=1  
 I IVTYPE="I" D  
 .I ADDCNT=0,SOLCNT>0 Q  
 .I ADDCNT=0 S RESULT=1 Q  
 .I ADDCNT'=STRCNT S RESULT=1 Q  
 Q RESULT  
 ;  
IVADFCHK(ORDIALOG) ;  
 ; This line tag checks to see if there are the same number of values  
 ;for ADDITIVE and Additive Frequency. This also checks to see if  
 ;the value assigned to ADDFREQ is one of the three possible values  
 ;All Bags, 1 bag/day, See Comments. If See Comments it also checks  
 ;for text in the comment section.  
 N ADDCNT,ADDFCNT,ADDFREQ,COMMENT,FREQ,INST,RESULT  
 S ADDCNT=$$VALCOUNT("ADDITIVE",.ORDIALOG)  
 S ADDFCNT=$$VALCOUNT("ADDITIVE FREQUENCY",.ORDIALOG)  
 I ADDCNT'=ADDFCNT Q 0  
 S ADDFREQ=$O(^ORD(101.41,"AB","OR GTX ADDITIVE FREQUENCY",""))  
 S COMMENT=$O(^ORD(101.41,"AB","OR GTX WORD PROCESSING 1",""))  
 I +$G(ADDFREQ)'>0 Q O  
 S INST=0,RESULT=1  
 F  S INST=$O(ORDIALOG(ADDFREQ,INST)) Q:INST'>0!(RESULT=0) D  
 .S FREQ=$$ADDFRQCV^ORMBLDP1($G(ORDIALOG(ADDFREQ,INST)),"O")  
 .I FREQ="A"!(FREQ=1) Q  
 .I FREQ="" S RESULT=0 Q  
 .I FREQ="S",'$L($G(ORDIALOG(COMMENT,1))) S RESULT=0 Q  
 Q RESULT  
 ;  
KEYVAR(DLG) ; Parse entry action for key variables & return in string  
 ; RV=CollTp^Samp^Spec^CollDt^Urg^Sched^NoComm^NoDiag^NoProv^NoRsn  
 N XCODE,RV,POS,Z  
 S XCODE=$G(^ORD(101.41,DLG,3)),RV=""  
 I '$L(XCODE) Q ""  
 S POS=$F(XCODE,"LRFZX=") I POS S $P(RV,U,1)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSAMP=") I POS S $P(RV,U,2)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSPEC=") I POS S $P(RV,U,3)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFDATE=") I POS S $P(RV,U,4)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFURG=") I POS S $P(RV,U,5)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"LRFSCH=") I POS S $P(RV,U,6)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"PSJNOPC=") I POS S $P(RV,U,7)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCNOPD=") I POS S $P(RV,U,8)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCNOAT=") I POS S $P(RV,U,9)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"GMRCREAF=") I POS S $P(RV,U,10)=$$VALUE(XCODE,POS)  
 S POS=$F(XCODE,"ORFORGET=") I POS D  
 . ; need to change this so that it is executed in SETKEYV so  
 . ; that it is executed each time menu is revisited  
 . N ORFORGET S ORFORGET=$$VALUE(XCODE,POS)  
 . I ORFORGET K ^TMP("ORECALL",$J,+ORFORGET)  
 . E  K ^TMP("ORECALL",$J)  
 Q RV  
VALUE(STR,BEG) ; Return value of "var=" (copied from ORCONVRT)  
 N X,Y,I S X=$E(STR,BEG,999),Y=""  
 S:$E(X)="""" X=$E(X,2,999) ; strip leading "  
 F I=1:1:$L(X) S Z=$E(X,I) Q:(Z=",")!(Z=" ")!(Z="""") S Y=Y\_Z  
 Q $TR(Y,U,"")  
 ;  
SETKEYV(X) ; Set the key variables based on contents of X  
 I $L($P(X,U,1)) S LRFZX=$P(X,U,1)  
 I $L($P(X,U,2)) S LRFSAMP=$P(X,U,2)  
 I $L($P(X,U,3)) S LRFSPEC=$P(X,U,3)  
 I $L($P(X,U,4)) S LRFDATE=$P(X,U,4)  
 I $L($P(X,U,5)) S LRFURG=$P(X,U,5)  
 I $L($P(X,U,6)) S LRFSCH=$P(X,U,6)  
 I $L($P(X,U,7)) S PSJNOPC=$P(X,U,7)  
 I $L($P(X,U,8)) S GMRCNOPD=$P(X,U,8)  
 I $L($P(X,U,9)) S GMRCNOAT=$P(X,U,9)  
 I $L($P(X,U,10)) S GMRCREAF=$P(X,U,10)  
 Q  
DLGINFO(IEN,MODE) ; return information about a dialog  
 ; IEN=DlgIEN or ORIFN, MODE=0:Dlg,1:Copy,2:Change  
 ; RESULT=DlgIEN^DlgType^FormID^DGrp  
 ; If MODE="1;T",don't check "PS MEDS" for transfer order  
 ; PSMDGP=1: Unit/Dose Group  
 ; PSMDGP=2: OutPatient Group  
 N X0,DLGIEN,TYP,FID,DGRP,PSMDGP,ISXF  
 S PSMDGP=0,ISXF=""  
 S ISXF=$P(MODE,";",2)  
 S MODE=+MODE  
 S DLGIEN=IEN I MODE,(ISXF'="T") D  
 . S DLGIEN=+$P($G(^OR(100,+IEN,0)),U,5)  
 . I $P(^ORD(101.41,DLGIEN,0),U)="PS MEDS" D  
 . . N PTCAT S PTCAT=$P($G(^OR(100,+IEN,0)),U,12)  
 . . I PTCAT="I" S DLGIEN=$O(^ORD(101.41,"B","PSJ OR PAT OE",0)),PSMDGP=1  
 . . I PTCAT="O" S DLGIEN=$O(^ORD(101.41,"B","PSO OERR",0)),PSMDGP=2  
 I MODE,(ISXF="T") S DLGIEN=+$P($G(^OR(100,+IEN,0)),U,5)  
 S X0=$G(^ORD(101.41,DLGIEN,0)),TYP=$P(X0,U,4),DGRP=$P(X0,U,5)  
 I MODE S DGRP=+$P($G(^OR(100,+IEN,0)),U,11)  
 ;JD NEW START 11/13/02  
 I DLGIEN=$O(^ORD(101.41,"B","PSJ OR PAT OE",0)) S PSMDGP=1  
 I DLGIEN=$O(^ORD(101.41,"B","PSO OERR",0)) S PSMDGP=2  
 ;JD NEW END 11/13/02  
 ; for copy or change, if the base dialog has changed, use it's info  
 I MODE,$G(ORDIALOG),(+DLGIEN'=+ORDIALOG),(PSMDGP=0) D  
 . S DLGIEN=+ORDIALOG,DGRP=$P(^ORD(101.41,+ORDIALOG,0),U,5)  
 D FORMID^ORWDXM(.FID,DLGIEN)  
 Q DLGIEN\_U\_TYP\_U\_FID\_U\_DGRP  
 ;  
CHKDSBL(LST,ID,MODE) ; return message if dialog disabled  
 ; ID=DlgIEN or ORIFN, MODE=0:Dialog,1:Copy,2:Change  
 ; LST=QL\_REJECT + disabled message or unchanged  
 S DLGIEN=+ID I MODE S DLGIEN=+$P($G(^OR(100,+ID,0)),U,5)  
 S X0=$G(^ORD(101.41,DLGIEN,0)),X=$P(X0,U,3)  
 I '$L(X),($P(X0,U,4)="Q") D  ; check default dialog  
 . S DLGIEN=+$$DEFDLG^ORWDXQ($P(X0,U,5))  
 . S X=$P($G(^ORD(101.41,DLGIEN,0)),U,3)  
 I $L(X) D  
 . I MODE D GETTXT^ORWORR(.LST,ID) S LST(.6)="",LST(.7)="Cannot "\_$S(MODE=1:"Copy",1:"Change")\_" -"  
 . S LST(0)="8^0",LST(.5)="Dialog Disabled: "\_X  
 Q  
CHKVACT(LST,ID,MODE,ORNP) ; return message if action not valid  
 ; ID=DlgIEN or ORIFN, MODE=0:Dialog,1:Copy,2:Change  
 ; LST=QL\_REJECT + invalid action message or unchanged  
 Q:'MODE  ; not an action on an order  
 N X,ACT S ACT=$S(MODE=1:"RW",MODE=2:"XX",1:"")  
 D VALID^ORWDXA(.X,ID,ACT,ORNP)  
 I $L(X) D GETTXT^ORWORR(.LST,ID) D  
 . S LST(0)="8^0",LST(.5)=X,LST(.6)="",LST(.7)="Cannot "\_$S(MODE=1:"Copy",1:"Change")\_" -"  
 Q  
CHKCOPY(LST,ID,FLDS) ; return message if can't copy this order  
 ; ID=ORIFN;ACT FLDS=EventType in 7th piece  
 ; LST=QL\_REJECT + cannot copy message or unchanged  
 I "^A^D^T^"'[(U\_$E($P(FLDS,U,7))\_U) Q             ; not event delayed  
 N PKG S PKG=$P($G(^OR(100,+ID,0)),U,14)  
 S PKG=$$NMSP^ORCD(PKG) I PKG="OR"!(PKG="PS") Q    ; xfer meds, generics  
 N ORWCAT S ORWCAT=$P($G(^OR(100,+ID,0)),U,12)  
 I ORWCAT="I",("^A^T^"[(U\_$E($P(FLDS,U,7))\_U)) Q   ; admit, xfer inpt  
 I ORWCAT="O",$E($P(FLDS,U,7))="D" Q               ; discharge outpt  
 D GETTXT^ORWORR(.LST,ID)  
 I ORWCAT="I" S LST(.5)="inpatient order to outpatient -"  
 I ORWCAT="O" S LST(.5)="outpatient order to inpatient -"  
 S:$D(LST(.5)) LST(.5)="Cannot copy the following "\_LST(.5)  
 S LST(0)="8^0",LST(.7)=""  
 Q  
BLD4CHG(LST,ID,FLDS) ; build responses for an edit  
 ; ID=ORIFN;ACT FLDS=unused right now  
 ; LST(0)=Qlvl^RespID(XOrderID)^DlgIEN^DlgType^FormID^DGrp  
 N OIDX,OI,CNT  
 S (OI,OIDX,CNT)=0  
 S:$D(^OR(100,+ID,4.5,"ID","ORDERABLE")) OIDX=$O(^OR(100,+ID,4.5,"ID","ORDERABLE",0))  
 I $D(^OR(100,+ID,4.5,OIDX)) D  
 . F  S CNT=$O(^OR(100,+ID,4.5,OIDX,CNT)) Q:'CNT  D  
 . . S OI=^(CNT) D VALDOI  
 I +LST(0)=8 S LST(.5)="You can not change this order." Q  
 S LST(0)="0^X"\_ID\_U\_$$DLGINFO(+ID,2)  
 S $P(LST(0),U,4)="X"  
 Q  
GETIVTYP() ;  
 N RESULT,TYPEIEN  
 S RESULT=""  
 S TYPEIEN=$O(^ORD(101.41,"B","OR GTX IV TYPE","")) I TYPEIEN'>0 Q RESULT  
 S RESULT=$G(ORDIALOG(TYPEIEN,1))  
 Q RESULT  
 ;  
ISTUBEQO(IFN) ;  
 N DG,DIAL  
 S DG=+$P($G(^ORD(101.41,IFN,0)),U,5)  
 S DIAL=$P($G(^ORD(100.98,DG,0)),U,4)  
 I +$G(DIAL)=0 Q 0  
 I $P($G(^ORD(101.41,DIAL,0)),U)="FHW8" Q 1  
 Q 0  
 ;  
VALDOI ; Validate the Orderable Items  
 N ORQUIT,ORPS  
 I $G(^ORD(101.43,OI,.1)),^(.1)'>$$NOW^XLFDT D  
 . S ORQUIT=1  
 . S LST(0)="8^0"  
 I $D(ORQUIT) Q:ORQUIT  
 S ORPS=$G(^ORD(101.43,+OI,"PS"))  
 I $P(ORPS,U,1,4)="0^0^0^0",($P(ORPS,U,7)=0) S LST(0)="8^0"  
 Q  
VERDUR(ORDIALOG) ;  
 ;check for duration value if a THEN conjunation is used  
 N CONJ,CONVALUE,DUR,I,SUCC  
 S SUCC=1  
 S CONJ=$$PTR^ORCDPS1("AND/THEN")  
 S DUR=$$PTR^ORCDPS1("DURATION")  
 S I=0 F  S I=$O(ORDIALOG(CONJ,I)) Q:I'>0!(SUCC=0) D  
 . I $$UP^XLFSTR($E($G(ORDIALOG(CONJ,I)),1))="T" D  
 . . I '$L($G(ORDIALOG(DUR,I))) S SUCC=0  
 Q SUCC  
 ;  
VERORD(OIEN) ;  
 N IFN,INFUSE,INFUID,ODG,ODP,ASSIV,SUCC,TYPE  
 S SUCC=0,IFN=ORDIALOG  
 S ODP=+$P($G(^ORD(101.41,+IFN,0)),U,7),ODG=+$P($G(^(0)),U,5)  
 S ODP=$$GET1^DIQ(9.4,+ODP\_",",1),ODG=$P($G(^ORD(100.98,ODG,0)),U,3)  
 I ODP'["PS" Q 1  
 I ODP="PSH" Q 1  
 ;check infusion rate for IV QO  
 I ODG="IV RX"!(ODG="TPN") D  Q SUCC  
 .S TYPE=$$GETIVTYP  
 .I TYPE="" Q  
 .S PASSIV=$$IVRTECHK  
 .I PASSIV=0 Q  
 .S INFUID=$O(^ORD(101.41,"B","OR GTX INFUSION RATE",0))  
 .S INFUSE=$G(ORDIALOG(INFUID,1))  
 .S SUCC=$$VALINF(TYPE,INFUSE)  
 .I SUCC=0 Q  
 .I TYPE="C" S SUCC=$$IVADFCHK(.ORDIALOG)  
 I (ODP="PSJ")!(ODP="PSO"),ODG'="IV RX",ODG'="TPN" S SUCC=$$VERDUR(.ORDIALOG)  
 Q SUCC  
 ;  
VALINF(TYPE,INFUSE) ;  
 N SUCC  
 S SUCC=0  
 I TYPE="I" D  Q SUCC  
 .I INFUSE="" S SUCC=1 Q  
 .I $TR(INFUSE,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLMNOPQRSTUVWXYZ")["INFUSE OVER" S SUCC=1 Q  
 .I $L(INFUSE)>4 Q  
 Q 1  
 ;  
VALQO(IFN) ;Check to see if it's a good QO med  
 ;If it's an IV QO: check if infusion rate entered  
 ;If it's an UD QO: check if dosage entered  
 ;regular order treated as good QO  
 ;  
 I IFN[";",($$UPCTCHK^ORWDXA(+IFN)) Q 0  
 I $P($G(^ORD(101.41,IFN,0)),U,4)'="Q" Q 1  
 N ODP,ODG,INFUID,INFUSE,DSAGEID,SUCC,PASSIV,TYPE  
 S SUCC=0  
 S ODP=+$P($G(^ORD(101.41,IFN,0)),U,7),ODG=+$P($G(^(0)),U,5)  
 S ODP=$$GET1^DIQ(9.4,+ODP\_",",1),ODG=$P($G(^ORD(100.98,ODG,0)),U,3)  
 I ODP'["PS" Q 1  
 I ODP="PSH" Q 1  
 ;check infusion rate for IV QO  
 I ODG="IV RX"!(ODG="TPN") D  
 . S INFUID=$O(^ORD(101.41,"B","OR GTX INFUSION RATE",0))  
 . S TYPE=$$GETIVTYP  
 . I TYPE="" Q  
 . I $D(ORDIALOG(INFUID,1)) D  
 . . I TYPE="I" D  Q  
 . . . S INFUSE=$G(ORDIALOG(INFUID,1))  
 . . . I INFUSE="" Q  
 . . . I INFUSE["INFUSE OVER" S SUCC=1 Q  
 . . . I $L(INFUSE)>4 Q  
 . . . I +INFUSE>0 S INFUSE="INFUSE OVER "\_INFUSE\_" Minutes"  
 . . . S ORDIALOG(INFUID,1)=INFUSE,SUCC=1  
 . . S SUCC=1  
 . ; additive frequency check/infusion rate checks for continuous orders  
 . I TYPE="C" D  I SUCC=0 Q  
 . . I $D(ORDIALOG(INFUID,1)) S SUCC=1 I SUCC=0 Q  
 . . S SUCC=$$IVADFCHK(.ORDIALOG)  
 . I SUCC=0 Q  
 . I '$D(ORDIALOG(INFUID,1)),TYPE="I" S SUCC=1  
 . S PASSIV=$$IVRTECHK  
 . I SUCC=0 Q  
 . I PASSIV=0 S SUCC=0  
 . I SUCC=1,$$ISMISSFL(.ORDIALOG,TYPE)=1 S SUCC=0  
 ;check dosage for UD QO  
 I (ODP="PSJ")!(ODP="PSO"),ODG'="IV RX",ODG'="TPN" D  
 . S DSAGEID=$O(^ORD(101.41,"B","OR GTX INSTRUCTIONS",0))  
 . I $D(ORDIALOG(DSAGEID,1)) S SUCC=1  
 . I SUCC=0 Q  
 . ;  
 . S SUCC=$$VERDUR(.ORDIALOG)  
 ;  
 I SUCC=1,$P($G(^ORD(101.41,IFN,5)),U,8) D  
 .N COMMID,WPCNT  
 .S COMMID=$O(^ORD(101.41,"B","OR GTX WORD PROCESSING 1",0))  
 .S COMMID=$O(^ORD(101.41,IFN,6,"D",COMMID,0))  
 .I COMMID S WPCNT=0 F  S WPCNT=$O(^ORD(101.41,IFN,6,COMMID,2,WPCNT)) Q:'WPCNT!('SUCC) D  
 ..I ^ORD(101.41,IFN,6,COMMID,2,WPCNT,0)["^" S SUCC=0  
 Q SUCC  
 ;  
IVRTECHK() ;  
 N RTIEN,RTVALUE,RESULT  
 N CNT,NUM,ORDERIDS,OIIEN,OTYPE,ROUTE  
 S CNT=0,RESULT=0  
 S RTIEN=+$P($G(ORDIALOG("B","ROUTE")),U,2) I RTIEN'>0 Q RESULT  
 S RTVALUE=+$G(ORDIALOG(RTIEN,1)) I RTVALUE'>0 Q RESULT  
 F OTYPE="SOLUTION","ADDITIVE" D  
 .S OIIEN=+$P($G(ORDIALOG("B",OTYPE)),U,2) I OIIEN>0 D  
 ..S NUM=0 F  S NUM=$O(ORDIALOG(OIIEN,NUM)) Q:NUM'>0 I +$G(ORDIALOG(OIIEN,NUM))>0 D  
 ...S CNT=CNT+1,ORDERIDS(CNT)=ORDIALOG(OIIEN,NUM)  
 I $D(ORDERIDS)=0 Q  
 S ROUTE=$$IVQOVAL^ORWDPS33(.ORDERIDS,RTVALUE)  
 I ROUTE="" S ORDIALOG(RTIEN,1)=ROUTE  
 I ROUTE'="" S RESULT=1  
 ;K ^TMP($J,"ORWDXM3 IVRTECHK")  
 ;D ALL^PSS51P2(RTVALUE,,,,"ORWDXM3 IVRTECHK")  
 ;I +^TMP($J,"ORWDXM3 IVRTECHK",RTVALUE,6)'=1 S ORDIALOG(RTIEN,1)="",RESULT=0  
 ;K ^TMP($J,"ORWDXM3 IVRTECHK")  
 Q RESULT  
 ;  
ISUDQO(ORY,DLGID) ;True: is unit dose quick order  
 S ORY=0  
 Q:'$D(^ORD(101.41,DLGID,0))  
 N CLODGRP,CLIVDGRP,UDGRP1,UDGRP2,DLGTYP,DLGGRP  
 S UDGRP1=$O(^ORD(100.98,"B","UD RX",0))  
 S UDGRP2=$O(^ORD(100.98,"B","I RX",0))  
 S CLODGRP=$O(^ORD(100.98,"B","CLINIC MEDICATIONS",""))  
 S CLIVDGRP=$O(^ORD(100.98,"B","CLINIC INFUSIONS",""))  
 S DLGTYP=$P($G(^ORD(101.41,DLGID,0)),U,4)  
 I DLGTYP="Q" S ^TMP("ZZ QUICK ORDER AUDIT",$J,"DLGID")=DLGID  
 S DLGGRP=$P($G(^ORD(101.41,DLGID,0)),U,5)  
 I (DLGTYP="Q"),((DLGGRP=UDGRP1)!(DLGGRP=UDGRP2)!(DLGGRP=CLODGRP)!(DLGGRP=CLIVDGRP)) S ORY=1  
 Q

**ORWPT**

**Before:**

ORWPT ; SLC/KCM/REV - Patient Lookup Functions ; 6/2/14 2:16pm  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*10,85,132,149,206,187,190,215,243,280,306,311,431,441\*\*;Dec 17, 1997;Build 30  
 ;  
 ; Ref. to ^UTILITY via IA 10061  
 ;  
IDINFO(REC,DFN) ; Return identifying information for a patient  
 ; PID^DOB^SEX^VET^SC%^WARD^RM-BED^NAME  
 N X0,X1,X101,X3,XV ; name/dob/sex/ssn, ward, room-bed, sc%, vet  
 S X0=$G(^DPT(DFN,0)),X1=$G(^(.1)),X101=$G(^(.101)),X3=$G(^(.3)),XV=$G(^("VET"))  
 S REC=$$SSN^DPTLK1(DFN)\_U\_$$DOB^DPTLK1(DFN,2)\_U\_$P(X0,U,2)\_U\_$P(XV,U)\_U\_$P(X3,U,2)\_U\_$P(X1,U)\_U\_$P(X101,U)\_U\_$P(X0,U) ;DG249  
 Q  
PTINQ(REF,DFN) ; Return formatted pt inquiry report  
 K ^TMP("ORDATA",$J,1)  
 D DGINQ^ORCXPND1(DFN)  
 S REF=$NA(^TMP("ORDATA",$J,1))  
 Q  
SCDIS(LST,DFN) ; Return service connected % and rated disabilities  
 N VAEL,VAERR,I,ILST,DIS,SC,X  
 D ELIG^VADPT  
 S LST(1)="Service Connected: "\_$S(+VAEL(3):$P(VAEL(3),U,2)\_"%",1:"NO")  
 I 'VAEL(4),'$P($G(^DG(391,+VAEL(6),0)),U,2) S LST(2)="NOT A VETERAN." Q  
 S I=0,ILST=1 F  S I=$O(^DPT(DFN,.372,I)) Q:'I  S X=^(I,0) D  
 . S DIS=$P($G(^DIC(31,+X,0)),U) Q:DIS=""  
 . S SC=$S($P(X,U,3):"SC",$P(X,U,3)']"":"not specified",1:"NSC")  
 . S ILST=ILST+1,LST(ILST)=DIS\_" ("\_$P(X,U,2)\_"% "\_SC\_")"  
 I ILST=1 S LST(2)="Rated Disabilities: NONE STATED"  
 Q  
SHOW ; temporary - show patient inquiry screen  
 N I,Y,DIC S DIC=2,DIC(0)="AEMQ" D ^DIC Q:'Y  
 K ^TMP("ORDATA",$J,1)  
 D DGINQ^ORCXPND1(+Y)  
 S I=0 F  S I=$O(^TMP("ORDATA",$J,1,I)) Q:'I  W !,^(I)  
 K ^TMP("ORDATA",$J,1)  
 Q  
SELCHK(REC,DFN) ; Check for sensitive pt  
 ; SENSITIVE  
 S REC=$$EN1^ORQPT2(DFN)  
 Q  
DIEDON(VAL,DFN) ; Check for a date of death  
 S VAL=+$G(^DPT(DFN,.35))  
 Q  
SELECT(REC,DFN) ; Selects patient & returns key information  
 ; 1 2 3 4 5 6 7 8 9 10 11 12  
 ; NAME^SEX^DOB^SSN^LOCIEN^LOCNM^RMBD^CWAD^SENSITIVE^ADMITTED^CONV^SC^  
 ; 13 14 15 16 17  
 ; SC%^ICN^AGE^TS^TSSVC  
 ;  
 ; for CCOW (RV - 2/27/03) name="-1", location=error message  
 I '$D(^DPT(+DFN,0)) S REC="-1^^^^^Patient is unknown to CPRS." Q  
 ;  
 N X  
 I $G(XWB("2","RPC"))="ORWPT SELECT" K ^TMP($J,"OC-OPOS") ; delete once per order session order checks  
 K ^TMP("ORWPCE",$J) ; delete PCE 'cache' when switching patients  
 S X=^DPT(DFN,0),REC=$P(X,U,1,3)\_U\_$P(X,U,9)\_U\_U\_$G(^(.1))\_U\_$G(^(.101))  
 S X=$P(REC,U,6) I $L(X) S $P(REC,U,5)=+$G(^DIC(42,+$O(^DIC(42,"B",X,0)),44))  
 S $P(REC,U,8)=$$CWAD^ORQPT2(DFN)\_U\_$$EN1^ORQPT2(DFN)  
 ; I $P(REC,U,9) D EN2^ORQPT2(DFN) ;update DG security log ; DG249  
 S X=$G(^DPT(DFN,.105)) I X S $P(REC,U,10)=$P($G(^DGPM(X,0)),U)  
 S:'$D(IOST) IOST="P-OTHER"  
 S $P(REC,U,11)=0  
 D ELIG^VADPT S $P(REC,U,12)=$G(VAEL(3)) ;two pieces: SC^SC%  
 I $L($T(GETICN^MPIF001)) S X=+$$GETICN^MPIF001(DFN) S:X>0 $P(REC,U,14)=X  
 S $P(REC,U,15)=$$AGE(DFN,$P(REC,U,3))  
 S $P(REC,U,16)=+$G(^DPT(DFN,.103)) ; treating specialty  
 I +$P(REC,U,16)>0 D  
 . N X,Y,Z  
 . S (X,Y)=""  
 . S X=$$TSDATA^DGACT(45.7,+$P(REC,U,16),.Y,"")  
 . I +X,+$P($G(Y(2)),U,1)>0 S (X,Z)="" S X=$$TSDATA^DGACT(42.4,+$P($G(Y(2)),U,1),.Z,"")  
 . I +X S $P(REC,U,17)=$P($G(Z(3)),U,1) ; treating specialty service  
 K VAEL,VAERR ;VADPT call to kill?  
 S ^DISV(DUZ,"^DPT(")=DFN  
 Q  
SHARE(VAL,IP,HWND,DFN) ; Set global to share DFN with other applications  
 K ^TMP("ORWCHART",$J),^TMP("ORECALL",$J),^TMP("ORWORD",$J)  
 K ^TMP("ORWDXMQ",$J)  
 S ^TMP("ORWCHART",$J,IP,HWND)=DFN  
 Q  
BYWARD(LST,WARD) ; Return a list of patients in a ward  
 N ILST,DFN  
 I +$G(WARD)<1 S LST(1)="^No ward identified" Q  
 S (ILST,DFN)=0  
 S WARD=$P(^DIC(42,WARD,0),"^") ;DBIA #36  
 F  S DFN=$O(^DPT("CN",WARD,DFN)) Q:DFN'>0 D  
 . S ILST=ILST+1,LST(ILST)=+DFN\_U\_$P(^DPT(+DFN,0),U)\_U\_$G(^DPT(+DFN,.101))  
 I ILST<1 S LST(1)="^No patients found."  
 Q  
LAST5(LST,ID) ; Return a list of patients matching A9999 identifiers  
 N I,IEN,XREF  
 S (I,IEN)=0,XREF=$S($L(ID)=5:"BS5",1:"BS")  
 F  S IEN=$O(^DPT(XREF,ID,IEN)) Q:'IEN  D  
 . S I=I+1,LST(I)=IEN\_U\_$P(^DPT(IEN,0),U)\_U\_$$DOB^DPTLK1(IEN,2)\_U\_$$SSN^DPTLK1(IEN) ; DG249  
 Q  
 ;  
LAST5RPL(LST,ID) ; ; Return list matching A9999 id's, but from RPL only.  
 N ORRPL,ORCNT,ORPT,ORPIEN  
 ; IA \_\_\_\_ allows read access to NEW PERSON file node 101:  
 S ORRPL=$G(^VA(200,DUZ,101))  
 S ORRPL=$P(ORRPL,U,2)  
 I (('ORRPL)!(ORRPL="")) S LST(0)="" Q  
 ;  
 S (ORCNT,ORPT)=0  
 F  S ORPT=$O(^OR(100.21,ORRPL,10,ORPT)) Q:'ORPT  D  
 .S ORPIEN=+$G(^OR(100.21,ORRPL,10,ORPT,0))  
 .I ((ORPIEN<0)!(ORPIEN="")) Q  
 .S ORCNT=ORCNT+1  
 .S LST(ORCNT)=ORPIEN\_U\_$P(^DPT(ORPIEN,0),U)\_U\_$$DOB^DPTLK1(ORPIEN,2)\_U\_$$SSN^DPTLK1(ORPIEN) ; DG249.  
 ;  
 Q  
 ;  
FULLSSN(LST,ID) ; Return a list of patients matching full SSN entered  
 N I,IEN  
 S (I,IEN)=0  
 F  S IEN=$O(^DPT("SSN",ID,IEN)) Q:'IEN  D  
 . S I=I+1,LST(I)=IEN\_U\_$P(^DPT(IEN,0),U)\_U\_$$DOB^DPTLK1(IEN,2)\_U\_$$SSN^DPTLK1(IEN) ; DG249  
 Q  
 ;  
FSSNRPL(LST,ID) ; Return list matching Full SSN, but from RPL only.  
 N ORRPL,ORCNT,ORPT,ORLPT,ORPIEN  
 ; IA \_\_\_\_ allows read access to NEW PERSON file node 101:  
 S ORRPL=$G(^VA(200,DUZ,101))  
 S ORRPL=$P(ORRPL,U,2)  
 I (('ORRPL)!(ORRPL="")) S LST(0)="" Q  
 ;  
 S (ORCNT,ORPT)=0  
 F  S ORPT=$O(^DPT("SSN",ID,ORPT)) Q:'ORPT  D  
 .S ORLPT=0  
 .F  S ORLPT=$O(^OR(100.21,ORRPL,10,ORLPT)) Q:'ORLPT  D  
 ..S ORPIEN=+$G(^OR(100.21,ORRPL,10,ORLPT,0))  
 ..I ((ORPIEN<0)!(ORPIEN="")) Q  
 ..I (ORPIEN'=ORPT) Q  
 ..S ORCNT=ORCNT+1  
 ..S LST(ORCNT)=ORPIEN\_U\_$P(^DPT(ORPIEN,0),U)\_U\_$$DOB^DPTLK1(ORPIEN,2)\_U\_$$SSN^DPTLK1(ORPIEN) ; DG249.  
 ;  
 Q  
 ;  
TOP(LST) ; Return top for all patients list (last selected for now)  
 N IEN  
 S IEN=$G(^DISV(DUZ,"^DPT("))  
 I IEN S LST(1)=IEN\_U\_$P($G(^DPT(IEN,0)),U)  
 Q  
ENCTITL(REC,DFN,LOC,PROV) ; Return external values for encounter  
 ; LOCNAME^LOCABBR^ROOMBED^PROVNAME  
 S $P(REC,U,1)=$P($G(^SC(+LOC,0)),U,1,2)  
 S $P(REC,U,3)=$P($G(^DPT(DFN,.101)),U)  
 S $P(REC,U,4)=$P($G(^VA(200,+PROV,0)),U)  
 S ^TMP("ZZ QUICK ORDER AUDIT",$J,"REC")=REC ; MPLS OR\*L102 8/1/06  
 Q  
LISTALL(Y,FROM,DIR) ; Return a bolus of patient names. From is either Name or IEN^Name.  
 N I,IEN,CNT,FROMIEN,ORIDNAME S CNT=44,I=0,FROMIEN=0  
 I $P(FROM,U,2)'="" S FROMIEN=$P(FROM,U,1),FROM=$O(^DPT("B",$P(FROM,U,2)),-DIR)  
 F  S FROM=$O(^DPT("B",FROM),DIR) Q:FROM=""  D  Q:I=CNT  
 . S IEN=FROMIEN,FROMIEN=0 F  S IEN=$O(^DPT("B",FROM,IEN)) Q:'IEN  D  Q:I=CNT  
 . . S ORIDNAME=""  
 . . S ORIDNAME=$G(^DPT(IEN,0)) ; Get zero node name.  
 . . ; S X1=$G(^DPT(IEN,.1))\_" "\_$G(^DPT(IEN,.101))  
 . . S I=I+1 S Y(I)=IEN\_U\_FROM\_U\_U\_U\_U\_$P(ORIDNAME,U) ;\_"^"\_X ; \_"^"\_X1 ;" ("\_X\_")"  
 Q  
APPTLST(LST,DFN) ; return a list of appointments  
 ; APPTTIME^LOCIEN^LOCNAME^EXTSTATUS  
 N ERR,ERRMSG,VASD,VAERR K ^UTILITY("VASD",$J) ;IA 10061  
 S VASD("F")=$$HTFM^XLFDT($H-30,1)  
 S VASD("T")=$$HTFM^XLFDT($H+1,1)\_".2359"  
 S VASD("W")="123456789"  
 D SDA^ORQRY01(.ERR,.ERRMSG)  
 I ERR K ^UTILITY("VASD",$J) K LST S LST(1)=ERRMSG Q  
 S I=0 F  S I=$O(^UTILITY("VASD",$J,I)) Q:'I  D  
 . S LST(I)=$P(^UTILITY("VASD",$J,I,"I"),U,1,2)\_U\_$P(^("E"),U,2,3)  
 K ^UTILITY("VASD",$J)  
 Q  
ADMITLST(LST,DFN) ; return a list of admissions  
 ; MOVETIME^LOCIEN^LOCNAME^TYPE  
 N TIM,MOV,X0,Y,MTIM,XTYP,XLOC,HLOC,ILST S ILST=0  
 S TIM="" F  S TIM=$O(^DGPM("ATID1",DFN,TIM)) Q:TIM'>0 D  
 . S MOV=0 F  S MOV=$O(^DGPM("ATID1",DFN,TIM,MOV)) Q:MOV'>0 D  
 . . N VSTR,TIUDA  
 . . S X0=$G(^DGPM(MOV,0)) I X0']"" Q  
 . . S MTIM=$P(X0,U)  
 . . S XTYP=$P($G(^DG(405.1,+$P(X0,U,4),0)),U,1)  
 . . S XLOC=$P($G(^DIC(42,+$P(X0,U,6),0)),U,1),HLOC=+$G(^(44))  
 . . S VSTR=HLOC\_";"\_MTIM\_";H",TIUDA=$$HASDS^TIULX(DFN,VSTR)  
 . . S ILST=ILST+1,LST(ILST)=MTIM\_U\_HLOC\_U\_XLOC\_U\_XTYP\_U\_MOV\_U\_TIUDA  
 Q  
CLINRNG(LST) ; return date ranges for clinic appointments  
 S LST(1)="T;T^Today"  
 S LST(2)="T+1;T+1^Tomorrow"  
 S LST(3)="T-1;T-1^Yesterday"  
 S LST(4)="T-7;T^Past Week"  
 S LST(5)="T-31;T^Past Month"  
 S LST(6)="S^Specify Date Range..."  
 Q  
 ;  
 N %,%H,X,SUNDAY,START  
 S LST(1)=DT\_";"\_DT\_"^Today",X=$$HTFM^XLFDT($H+1,1)  
 S LST(2)=X\_";"\_X\_"^Tomorrow"  
 S X=+$H F  Q:X#7=3 S X=X-1 ; $H#7=3 is Sunday  
 S LST(3)=$$HTFM^XLFDT(X)\_";"\_$$HTFM^XLFDT(X+6)\_"^This Week"  
 S LST(4)=$$HTFM^XLFDT(X+7)\_";"\_$$HTFM^XLFDT(X+13)\_"^Next Week"  
 S LST(5)=$E(DT,1,5)\_"01;"\_$E(DT,1,5)\_"31^This Month"  
 S X=$E(DT,4,5)+1 S:X=13 X=1 S X=$E(DT,1,3)\_$TR($J(X,2)," ",0)  
 S LST(6)=X\_"01;"\_X\_"31^Next Month"  
 S LST(7)="^Specify Dates"  
 Q  
DFLTSRC(VAL) ; return default patient list source (T, W, C, P, S)  
 N SRV S SRV=+$G(^VA(200,DUZ,5))  
 S VAL=$$GET^XPAR("ALL^SRV.`"\_SRV,"ORLP DEFAULT LIST SOURCE")  
 Q  
SAVDFLT(OK,X) ; save new default patient list settings (X=type^ien^sdt;edt)  
 G SAVDFLT^ORWPT1  
 ;  
DISCHRG(Y,DFN,ADMITDT) ; Get discharge movement information  
 N VAIP  
 I +$G(ADMITDT)=0 S Y=DT Q  
 S VAIP("D")=ADMITDT D 52^VADPT  
 I +VAIP(17)=0 S Y=DT Q  
 S Y=+VAIP(17,1)  
 Q  
CWAD(Y,DFN) ; returns CWAD flags for a patient  
 S Y=$$CWAD^ORQPT2(DFN)  
 Q  
LEGACY(ORLST,DFN) ; return message if data on the legacy system  
 ; ORLST(0)=1 if data, ORLST(n)=display message if data  
 S ORLST(0)=0  
 I $L($T(HXDATA^A7RDPAGU)) D

. D HXDATA^A7RDPAGU(.ORLST,DFN)

. I $O(ORLST(0)) S ORLST(0)=1 Q  
INPLOC(REC,DFN) ; Return a patient's current location  
 N X  
 S X=$G(^DPT(DFN,.102)),REC=0  
 I X S X=$P($G(^DGPM(X,0)),U,6)  
 I X S REC=+$G(^DIC(42,X,44))  
 I X S $P(REC,U,2)=$P($G(^DIC(42,X,0)),U,1)  
 I X S X=$P($G(^DIC(42,X,0)),U,3)  
 S $P(REC,U,3)=X  
 Q  
AGE(DFN,BEG) ; returns age based on date of birth and date of death (or DT)  
 N END,X  
 S END=+$G(^DPT(DFN,.35)),END=$S(END:END,1:DT)  
 S X=$E(END,1,3)-$E(BEG,1,3)-($E(END,4,7)<$E(BEG,4,7))  
 Q X  
ROK(X) ; Routine OK (in UCI) (NDBI)  
 S X=$G(X) Q:'$L(X) 0 Q:$L(X)>8 0 X ^%ZOSF("TEST") Q:$T 1 Q 0  
 ;-  
 ;NDBI(X) ; National Database Integration site 1 = yes 0 = no  
 ; N R,G S X="A7RDUP" X ^%ZOSF("TEST") S R=$T,G=$S($D(^A7RCP):1,1:0),X=R+G,X=$S(X=2:1,1:0) Q X

**After:**

ORWPT ; SLC/KCM/REV - Patient Lookup Functions ; 6/2/14 2:16pm  
 ;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*10,85,132,149,206,187,190,215,243,280,306,311,L102\*\*;Dec 17, 1997;Build 30  
 ;  
 ; Ref. to ^UTILITY via IA 10061  
 ;  
IDINFO(REC,DFN) ; Return identifying information for a patient  
 ; PID^DOB^SEX^VET^SC%^WARD^RM-BED^NAME  
 N X0,X1,X101,X3,XV ; name/dob/sex/ssn, ward, room-bed, sc%, vet  
 S X0=$G(^DPT(DFN,0)),X1=$G(^(.1)),X101=$G(^(.101)),X3=$G(^(.3)),XV=$G(^("VET"))  
 S REC=$$SSN^DPTLK1(DFN)\_U\_$$DOB^DPTLK1(DFN,2)\_U\_$P(X0,U,2)\_U\_$P(XV,U)\_U\_$P(X3,U,2)\_U\_$P(X1,U)\_U\_$P(X101,U)\_U\_$P(X0,U) ;DG249  
 Q  
PTINQ(REF,DFN) ; Return formatted pt inquiry report  
 K ^TMP("ORDATA",$J,1)  
 D DGINQ^ORCXPND1(DFN)  
 S REF=$NA(^TMP("ORDATA",$J,1))  
 Q  
SCDIS(LST,DFN) ; Return service connected % and rated disabilities  
 N VAEL,VAERR,I,ILST,DIS,SC,X  
 D ELIG^VADPT  
 S LST(1)="Service Connected: "\_$S(+VAEL(3):$P(VAEL(3),U,2)\_"%",1:"NO")  
 I 'VAEL(4),'$P($G(^DG(391,+VAEL(6),0)),U,2) S LST(2)="NOT A VETERAN." Q  
 S I=0,ILST=1 F  S I=$O(^DPT(DFN,.372,I)) Q:'I  S X=^(I,0) D  
 . S DIS=$P($G(^DIC(31,+X,0)),U) Q:DIS=""  
 . S SC=$S($P(X,U,3):"SC",$P(X,U,3)']"":"not specified",1:"NSC")  
 . S ILST=ILST+1,LST(ILST)=DIS\_" ("\_$P(X,U,2)\_"% "\_SC\_")"  
 I ILST=1 S LST(2)="Rated Disabilities: NONE STATED"  
 Q  
SHOW ; temporary - show patient inquiry screen  
 N I,Y,DIC S DIC=2,DIC(0)="AEMQ" D ^DIC Q:'Y  
 K ^TMP("ORDATA",$J,1)  
 D DGINQ^ORCXPND1(+Y)  
 S I=0 F  S I=$O(^TMP("ORDATA",$J,1,I)) Q:'I  W !,^(I)  
 K ^TMP("ORDATA",$J,1)  
 Q  
SELCHK(REC,DFN) ; Check for sensitive pt  
 ; SENSITIVE  
 S REC=$$EN1^ORQPT2(DFN)  
 Q  
DIEDON(VAL,DFN) ; Check for a date of death  
 S VAL=+$G(^DPT(DFN,.35))  
 Q  
SELECT(REC,DFN) ; Selects patient & returns key information  
 ; 1 2 3 4 5 6 7 8 9 10 11 12  
 ; NAME^SEX^DOB^SSN^LOCIEN^LOCNM^RMBD^CWAD^SENSITIVE^ADMITTED^CONV^SC^  
 ; 13 14 15 16 17  
 ; SC%^ICN^AGE^TS^TSSVC  
 ;  
 ; for CCOW (RV - 2/27/03) name="-1", location=error message  
 I '$D(^DPT(+DFN,0)) S REC="-1^^^^^Patient is unknown to CPRS." Q  
 ;  
 N X  
 I $G(XWB("2","RPC"))="ORWPT SELECT" K ^TMP($J,"OC-OPOS") ; delete once per order session order checks  
 K ^TMP("ORWPCE",$J) ; delete PCE 'cache' when switching patients  
 S X=^DPT(DFN,0),REC=$P(X,U,1,3)\_U\_$P(X,U,9)\_U\_U\_$G(^(.1))\_U\_$G(^(.101))  
 S X=$P(REC,U,6) I $L(X) S $P(REC,U,5)=+$G(^DIC(42,+$O(^DIC(42,"B",X,0)),44))  
 S $P(REC,U,8)=$$CWAD^ORQPT2(DFN)\_U\_$$EN1^ORQPT2(DFN)  
 ; I $P(REC,U,9) D EN2^ORQPT2(DFN) ;update DG security log ; DG249  
 S X=$G(^DPT(DFN,.105)) I X S $P(REC,U,10)=$P($G(^DGPM(X,0)),U)  
 S:'$D(IOST) IOST="P-OTHER"  
 S $P(REC,U,11)=0  
 D ELIG^VADPT S $P(REC,U,12)=$G(VAEL(3)) ;two pieces: SC^SC%  
 I $L($T(GETICN^MPIF001)) S X=+$$GETICN^MPIF001(DFN) S:X>0 $P(REC,U,14)=X  
 S $P(REC,U,15)=$$AGE(DFN,$P(REC,U,3))  
 S $P(REC,U,16)=+$G(^DPT(DFN,.103)) ; treating specialty  
 I +$P(REC,U,16)>0 D  
 . N X,Y,Z  
 . S (X,Y)=""  
 . S X=$$TSDATA^DGACT(45.7,+$P(REC,U,16),.Y,"")  
 . I +X,+$P($G(Y(2)),U,1)>0 S (X,Z)="" S X=$$TSDATA^DGACT(42.4,+$P($G(Y(2)),U,1),.Z,"")  
 . I +X S $P(REC,U,17)=$P($G(Z(3)),U,1) ; treating specialty service  
 K VAEL,VAERR ;VADPT call to kill?  
 S ^DISV(DUZ,"^DPT(")=DFN  
 Q  
SHARE(VAL,IP,HWND,DFN) ; Set global to share DFN with other applications  
 K ^TMP("ORWCHART",$J),^TMP("ORECALL",$J),^TMP("ORWORD",$J)  
 K ^TMP("ORWDXMQ",$J)  
 S ^TMP("ORWCHART",$J,IP,HWND)=DFN  
 Q  
BYWARD(LST,WARD) ; Return a list of patients in a ward  
 N ILST,DFN  
 I +$G(WARD)<1 S LST(1)="^No ward identified" Q  
 S (ILST,DFN)=0  
 S WARD=$P(^DIC(42,WARD,0),"^") ;DBIA #36  
 F  S DFN=$O(^DPT("CN",WARD,DFN)) Q:DFN'>0 D  
 . S ILST=ILST+1,LST(ILST)=+DFN\_U\_$P(^DPT(+DFN,0),U)\_U\_$G(^DPT(+DFN,.101))  
 I ILST<1 S LST(1)="^No patients found."  
 Q  
LAST5(LST,ID) ; Return a list of patients matching A9999 identifiers  
 N I,IEN,XREF  
 S (I,IEN)=0,XREF=$S($L(ID)=5:"BS5",1:"BS")  
 F  S IEN=$O(^DPT(XREF,ID,IEN)) Q:'IEN  D  
 . S I=I+1,LST(I)=IEN\_U\_$P(^DPT(IEN,0),U)\_U\_$$DOB^DPTLK1(IEN,2)\_U\_$$SSN^DPTLK1(IEN) ; DG249  
 Q  
 ;  
LAST5RPL(LST,ID) ; ; Return list matching A9999 id's, but from RPL only.  
 N ORRPL,ORCNT,ORPT,ORPIEN  
 ; IA \_\_\_\_ allows read access to NEW PERSON file node 101:  
 S ORRPL=$G(^VA(200,DUZ,101))  
 S ORRPL=$P(ORRPL,U,2)  
 I (('ORRPL)!(ORRPL="")) S LST(0)="" Q  
 ;  
 S (ORCNT,ORPT)=0  
 F  S ORPT=$O(^OR(100.21,ORRPL,10,ORPT)) Q:'ORPT  D  
 .S ORPIEN=+$G(^OR(100.21,ORRPL,10,ORPT,0))  
 .I ((ORPIEN<0)!(ORPIEN="")) Q  
 .S ORCNT=ORCNT+1  
 .S LST(ORCNT)=ORPIEN\_U\_$P(^DPT(ORPIEN,0),U)\_U\_$$DOB^DPTLK1(ORPIEN,2)\_U\_$$SSN^DPTLK1(ORPIEN) ; DG249.  
 ;  
 Q  
 ;  
FULLSSN(LST,ID) ; Return a list of patients matching full SSN entered  
 N I,IEN  
 S (I,IEN)=0  
 F  S IEN=$O(^DPT("SSN",ID,IEN)) Q:'IEN  D  
 . S I=I+1,LST(I)=IEN\_U\_$P(^DPT(IEN,0),U)\_U\_$$DOB^DPTLK1(IEN,2)\_U\_$$SSN^DPTLK1(IEN) ; DG249  
 Q  
 ;  
FSSNRPL(LST,ID) ; Return list matching Full SSN, but from RPL only.  
 N ORRPL,ORCNT,ORPT,ORLPT,ORPIEN  
 ; IA \_\_\_\_ allows read access to NEW PERSON file node 101:  
 S ORRPL=$G(^VA(200,DUZ,101))  
 S ORRPL=$P(ORRPL,U,2)  
 I (('ORRPL)!(ORRPL="")) S LST(0)="" Q  
 ;  
 S (ORCNT,ORPT)=0  
 F  S ORPT=$O(^DPT("SSN",ID,ORPT)) Q:'ORPT  D  
 .S ORLPT=0  
 .F  S ORLPT=$O(^OR(100.21,ORRPL,10,ORLPT)) Q:'ORLPT  D  
 ..S ORPIEN=+$G(^OR(100.21,ORRPL,10,ORLPT,0))  
 ..I ((ORPIEN<0)!(ORPIEN="")) Q  
 ..I (ORPIEN'=ORPT) Q  
 ..S ORCNT=ORCNT+1  
 ..S LST(ORCNT)=ORPIEN\_U\_$P(^DPT(ORPIEN,0),U)\_U\_$$DOB^DPTLK1(ORPIEN,2)\_U\_$$SSN^DPTLK1(ORPIEN) ; DG249.  
 ;  
 Q  
 ;  
TOP(LST) ; Return top for all patients list (last selected for now)  
 N IEN  
 S IEN=$G(^DISV(DUZ,"^DPT("))  
 I IEN S LST(1)=IEN\_U\_$P($G(^DPT(IEN,0)),U)  
 Q  
ENCTITL(REC,DFN,LOC,PROV) ; Return external values for encounter  
 ; LOCNAME^LOCABBR^ROOMBED^PROVNAME  
 S $P(REC,U,1)=$P($G(^SC(+LOC,0)),U,1,2)  
 S $P(REC,U,3)=$P($G(^DPT(DFN,.101)),U)  
 S $P(REC,U,4)=$P($G(^VA(200,+PROV,0)),U)  
 S ^TMP("ZZ QUICK ORDER AUDIT",$J,"REC")=REC ; MPLS OR\*L102 8/1/06  
 Q  
LISTALL(Y,FROM,DIR) ; Return a bolus of patient names. From is either Name or IEN^Name.  
 N I,IEN,CNT,FROMIEN,ORIDNAME S CNT=44,I=0,FROMIEN=0  
 I $P(FROM,U,2)'="" S FROMIEN=$P(FROM,U,1),FROM=$O(^DPT("B",$P(FROM,U,2)),-DIR)  
 F  S FROM=$O(^DPT("B",FROM),DIR) Q:FROM=""  D  Q:I=CNT  
 . S IEN=FROMIEN,FROMIEN=0 F  S IEN=$O(^DPT("B",FROM,IEN)) Q:'IEN  D  Q:I=CNT  
 . . S ORIDNAME=""  
 . . S ORIDNAME=$G(^DPT(IEN,0)) ; Get zero node name.  
 . . ; S X1=$G(^DPT(IEN,.1))\_" "\_$G(^DPT(IEN,.101))  
 . . S I=I+1 S Y(I)=IEN\_U\_FROM\_U\_U\_U\_U\_$P(ORIDNAME,U) ;\_"^"\_X ; \_"^"\_X1 ;" ("\_X\_")"  
 Q  
APPTLST(LST,DFN) ; return a list of appointments  
 ; APPTTIME^LOCIEN^LOCNAME^EXTSTATUS  
 N ERR,ERRMSG,VASD,VAERR K ^UTILITY("VASD",$J) ;IA 10061  
 S VASD("F")=$$HTFM^XLFDT($H-30,1)  
 S VASD("T")=$$HTFM^XLFDT($H+1,1)\_".2359"  
 S VASD("W")="123456789"  
 D SDA^ORQRY01(.ERR,.ERRMSG)  
 I ERR K ^UTILITY("VASD",$J) K LST S LST(1)=ERRMSG Q  
 S I=0 F  S I=$O(^UTILITY("VASD",$J,I)) Q:'I  D  
 . S LST(I)=$P(^UTILITY("VASD",$J,I,"I"),U,1,2)\_U\_$P(^("E"),U,2,3)  
 K ^UTILITY("VASD",$J)  
 Q  
ADMITLST(LST,DFN) ; return a list of admissions  
 ; MOVETIME^LOCIEN^LOCNAME^TYPE  
 N TIM,MOV,X0,Y,MTIM,XTYP,XLOC,HLOC,ILST S ILST=0  
 S TIM="" F  S TIM=$O(^DGPM("ATID1",DFN,TIM)) Q:TIM'>0 D  
 . S MOV=0 F  S MOV=$O(^DGPM("ATID1",DFN,TIM,MOV)) Q:MOV'>0 D  
 . . N VSTR,TIUDA  
 . . S X0=$G(^DGPM(MOV,0)) I X0']"" Q  
 . . S MTIM=$P(X0,U)  
 . . S XTYP=$P($G(^DG(405.1,+$P(X0,U,4),0)),U,1)  
 . . S XLOC=$P($G(^DIC(42,+$P(X0,U,6),0)),U,1),HLOC=+$G(^(44))  
 . . S VSTR=HLOC\_";"\_MTIM\_";H",TIUDA=$$HASDS^TIULX(DFN,VSTR)  
 . . S ILST=ILST+1,LST(ILST)=MTIM\_U\_HLOC\_U\_XLOC\_U\_XTYP\_U\_MOV\_U\_TIUDA  
 Q  
CLINRNG(LST) ; return date ranges for clinic appointments  
 S LST(1)="T;T^Today"  
 S LST(2)="T+1;T+1^Tomorrow"  
 S LST(3)="T-1;T-1^Yesterday"  
 S LST(4)="T-7;T^Past Week"  
 S LST(5)="T-31;T^Past Month"  
 S LST(6)="S^Specify Date Range..."  
 Q  
 ;  
 N %,%H,X,SUNDAY,START  
 S LST(1)=DT\_";"\_DT\_"^Today",X=$$HTFM^XLFDT($H+1,1)  
 S LST(2)=X\_";"\_X\_"^Tomorrow"  
 S X=+$H F  Q:X#7=3 S X=X-1 ; $H#7=3 is Sunday  
 S LST(3)=$$HTFM^XLFDT(X)\_";"\_$$HTFM^XLFDT(X+6)\_"^This Week"  
 S LST(4)=$$HTFM^XLFDT(X+7)\_";"\_$$HTFM^XLFDT(X+13)\_"^Next Week"  
 S LST(5)=$E(DT,1,5)\_"01;"\_$E(DT,1,5)\_"31^This Month"  
 S X=$E(DT,4,5)+1 S:X=13 X=1 S X=$E(DT,1,3)\_$TR($J(X,2)," ",0)  
 S LST(6)=X\_"01;"\_X\_"31^Next Month"  
 S LST(7)="^Specify Dates"  
 Q  
DFLTSRC(VAL) ; return default patient list source (T, W, C, P, S)  
 N SRV S SRV=+$G(^VA(200,DUZ,5))  
 S VAL=$$GET^XPAR("ALL^SRV.`"\_SRV,"ORLP DEFAULT LIST SOURCE")  
 Q  
SAVDFLT(OK,X) ; save new default patient list settings (X=type^ien^sdt;edt)  
 G SAVDFLT^ORWPT1  
 ;  
DISCHRG(Y,DFN,ADMITDT) ; Get discharge movement information  
 N VAIP  
 I +$G(ADMITDT)=0 S Y=DT Q  
 S VAIP("D")=ADMITDT D 52^VADPT  
 I +VAIP(17)=0 S Y=DT Q  
 S Y=+VAIP(17,1)  
 Q  
CWAD(Y,DFN) ; returns CWAD flags for a patient  
 S Y=$$CWAD^ORQPT2(DFN)  
 Q  
LEGACY(ORLST,DFN) ; return message if data on the legacy system  
 ; ORLST(0)=1 if data, ORLST(n)=display message if data  
 S ORLST(0)=0  
 I $L($T(HXDATA^ORPO7GUI)) D  
 . D HXDATA^ORPO7GUI(.ORLST,DFN)  
 . I $O(ORLST(0)) S ORLST(0)=1  
 Q  
INPLOC(REC,DFN) ; Return a patient's current location  
 N X  
 S X=$G(^DPT(DFN,.102)),REC=0  
 I X S X=$P($G(^DGPM(X,0)),U,6)  
 I X S REC=+$G(^DIC(42,X,44))  
 I X S $P(REC,U,2)=$P($G(^DIC(42,X,0)),U,1)  
 I X S X=$P($G(^DIC(42,X,0)),U,3)  
 S $P(REC,U,3)=X  
 Q  
AGE(DFN,BEG) ; returns age based on date of birth and date of death (or DT)  
 N END,X  
 S END=+$G(^DPT(DFN,.35)),END=$S(END:END,1:DT)  
 S X=$E(END,1,3)-$E(BEG,1,3)-($E(END,4,7)<$E(BEG,4,7))  
 Q X  
ROK(X) ; Routine OK (in UCI) (NDBI)  
 S X=$G(X) Q:'$L(X) 0 Q:$L(X)>8 0 X ^%ZOSF("TEST") Q:$T 1 Q 0  
 ;-  
 ;NDBI(X) ; National Database Integration site 1 = yes 0 = no  
 ; N R,G S X="A7RDUP" X ^%ZOSF("TEST") S R=$T,G=$S($D(^A7RCP):1,1:0),X=R+G,X=$S(X=2:1,1:0) Q X

**ORQOAUIA (New)**

ORQOAUIA ;ALB/RTW - DAILY TASK RETRIEVE ASSOCIATED ORDER FOR A QUICK ORDER ; 11/20/16 8:31pm  
 ;;1.1;ORQOA QUICK ORDER AUDIT V;\*\*405\*\*;Jun 08, 2015  
 Q  
START ; ENTRY POINT  
 S ORDAT=$$NOW^XLFDT\1 ;PROCESS RECORDS UP TO 7 DAYS OLD  
 F  S ORDAT=$O(^ANAZ(6189050,"B",ORDAT)) Q:+ORDAT'>0 S ORI=0 F  S ORI=$O(^ANAZ(6189050,"B",ORDAT,ORI)) Q:+ORI'>0 D  
 . S ORX0=^ANAZ(6189050,ORI,0),ORQOIFN=$P(ORX0,U,4)  
 . S ORD1=$O(^ORD(101.41,ORQOIFN,6,"D",4,0))  
 . Q:ORD1=""  ;QUIT IF NO ORDERABLE ITEM SPECIFIED ; NEW LINE 2/7/11  
 . S ORDITEM=+^ORD(101.41,ORQOIFN,6,ORD1,1) ; IFN OF ORDERABLE ITEM  
 . S ORDFN=$P(ORX0,"^",3),ORDFNPLS=ORDFN\_";DPT("  
 . S (ORDAT2,ORSTOP)=0 F  S ORDAT2=$O(^OR(100,"ACT",ORDFNPLS,ORDAT2)) Q:+ORDAT2'>0!(ORSTOP) S ORJ=0 F  S ORJ=$O(^OR(100,"AC",ORDFNPLS,ORDAT2,ORJ)) Q:+ORJ'>0 D  
 .. S ORDATORD=$P(^OR(100,ORJ,0),"^",7)+.000099 ;DATE/TIME ORDER ENTERED  
 .. S ORDTQUIC=$P(^ANAZ(6189050,ORI,0),"^",1) ;DATE/TIME OF AUDIT RECORD  
 .. I ORDATORD<ORDTQUIC S ORSTOP=1 ;CHECKING OLDER ORDERS, ORSTOP CHECKING  
 .. I ORDATORD>ORDTQUIC,(ORDATORD\1)=(ORDTQUIC\1) D  ;SAME DATE & GREATER TIME  
 ... ; MATCH ORFND, STORE IFN OF ORDER IN AUDIT FILE  
 ... I $D(^OR(100,ORJ,.1,"B",ORDITEM)) S $P(^ANAZ(6189050,ORI,0),"^",8)=ORJ,ORSTOP=1  
 Q  
START3 ; ENTRY POINT 2  
 N ORDAT,ORDAT2,ORDATORD,ORDTQUIC,ORDFN,ORDFNPLS,ORFND,ORI,ORJ,ORJJ,ORJO,ORK,ORKK,ORKK0,ORDITEM,ORQOIFN,ORREPLCD,ORSTOP,ORX,ORX0,ORD1  
 S ORDAT=$$FMADD^XLFDT(DT,-30) ; PROCESS RECORDS UP TO 30 DAYS OLD  
 I '$D(ZTQUEUED) W !,"Audit#",?10,"Order",?20,"Comment"  
 F  S ORDAT=$O(^ANAZ(6189050,"B",ORDAT)) Q:+ORDAT'>0 S ORI=0 F  S ORI=$O(^ANAZ(6189050,"B",ORDAT,ORI)) Q:+ORI'>0 D  
 . S ORX0=^ANAZ(6189050,ORI,0),ORQOIFN=$P(ORX0,U,4)  
 . Q:$P(ORX0,U,8)>0 ;Quit if entry already has a ORDER NUMBER.  
 . Q:ORQOIFN=""  ;NO QUICK ORDER RECORDED  
 . S ORD1=$O(^ORD(101.41,ORQOIFN,6,"D",4,0))  
 . S ORDFN=$P(ORX0,U,3),ORDFNPLS=ORDFN\_";DPT("  
 . S (ORDAT2,ORSTOP)=0 F  S ORDAT2=$O(^OR(100,"ACT",ORDFNPLS,ORDAT2)) Q:+ORDAT2'>0!(ORSTOP) S ORJJ=0 F  S ORJJ=$O(^OR(100,"ACT",ORDFNPLS,ORDAT2,ORJJ)) Q:+ORJJ'>0 D  
 .. S ORJ=0 F  S ORJ=$O(^OR(100,"ACT",ORDFNPLS,ORDAT2,ORJJ,ORJ)) Q:+ORJ'>0!(ORSTOP) D  
 ... S ORDATORD=$P(^OR(100,ORJ,0),"^",7)+.000099 ;DATE/TIME ORDER ENTERED  
 ... S ORDTQUIC=$P(^ANAZ(6189050,ORI,0),"^",1) ;DATE/TIM OFAUDIT RECORD  
 ... S ORREPLCD=$P($G(^OR(100,ORJ,3)),U,5) ;REPLACED ORDER number if any. ... S ORJO=ORJ D SCAN I 'ORFND,ORREPLCD>0 S ORJO=ORREPLCD D SCAN  
 Q  
SCAN ;  
 ;LOOK FOR AUDIT IFN IN COMMENTS, STORE IFN OF ORDER IN AUDIT FILE  
 S ORFND=0  
 ;  
 I ORKK0]"",ORKK0[("\*\* Pharmacy Confirmation #: "\_ORI) S $P(^ANAZ(6189050,ORI,0),"^",8)=ORJ,ORSTOP=1 I '$D(ZTQUEUED) W !,ORI,?10,ORJ,?20,ORKK0  
 I $D(^OR(100,ORJO,8,0)) S ORK=0 F  S ORK=$O(^OR(100,ORJO,8,ORK)) Q:+ORK'>0 I $D(^OR(100,ORJO,8,ORK,.1,0)) S ORKK=0 F  S ORKK=$O(^OR(100,ORJO,8,ORK,.1,ORKK)) Q:ORKK'>0 D  
 . S ORKK0=^OR(100,ORJO,8,ORK,.1,ORKK,0) I ORKK0["\*\* Pharmacy Confirmation",ORKK0[ORI S $P(^ANAZ(6189050,ORI,0),U,8)=ORJO,ORSTOP=1,ORFND=1 I '$D(ZTQUEUED) W !,ORI\_U\_ORDAT,!,?10,ORJ,?20,ORKK0  
 Q  
 ;

**ORQOAUIB (New)**

ORQOAUIB ;ALB/RTW - LIST ANTI-MICROBIAL ORDER, ORCDSS & NON-ORCDSS ; 11/23/16 7:45pm  
 ;;1.1;ORQOA QUICK ORDER AUDIT V;;Jun 08, 2015  
 N ORSDATE,OREDATE,ORI,ORJ,ORCDSS,ORDIV,ORI1,OR2,Y,X2,ORWHO,%DT,%IS,DIC,ZTIO,ZTRTN,ZTSAVE,ZTSK  
 K ^TMP($J)  
START S %DT="AE",%DT("A")="Enter Starting Date: " D ^%DT Q:+Y'>0 S ORSDATE=+Y  
 S %DT="AE",%DT("A")="Enter Ending Date: " D ^%DT Q:+Y'>0 S OREDATE=+Y  
 I OREDATE<ORSDATE W !,"\*\* ERROR \*\* - Ending Date needs to follow Starting Date",! G START  
 S ORDIV="" I $O(^OR(100.953,0))>0 S DIC(0)="AEQM",DIC="^OR(100.953,",DIC("A")="Select DIVISION GROUP: " D ^DIC Q:+Y'>0   
   
 S ORDIV=+Y  
 I $$FMDIFF^XLFDT(OREDATE,ORSDATE,1)>30 W !,"\*\* ERROR \*\* - Only 30 days at a time is permitted",! G START  
 S %IS="MQ" D ^%ZIS  
 I $D(IO("Q")) S ZTRTN="DQ^R2QOAUI5",ZTSAVE("ORSDATE")="",ZTSAVE("OREDATE")="",ZTIO=ION,ZTSAVE("IO\*")="",ZTSAVE("ORDIV")="" D ^%ZTLOAD W !,"Task Queued: ",ZTSK Q  
 U IO  
DQ ; QUEUED ENTRY POINT  
 S I=ORSDATE F  S ORI=$O(^OR(100,"AF",ORI)) Q:+ORI'>0!(+ORI>OREDATE) S ORJ=0 F  S ORJ=$O(^OR(100,"AF",ORI,ORJ)) Q:+ORJ'>0 D  
 . Q:'$$DIVMATCH(ORJ) ; QUIT IF NOT FOR ONE OF REQUESTED DIVISIONS  
 . Q:'$$ANTIMIC(ORJ) ; QUIT IF NOT ANTI-MICROBIAL ORDER  
 . S ORCDSS=$$ORCDSS(ORJ) ; DETERMINE WHETHER ORCDSS OR NON-ORCDSS ORDER  
 . I ORCDSS S ^TMP($J,"ORCDSS",ORJ)=ORCDSS  
 . E  S ^TMP($J,"NON-ORCDSS",ORJ)=""  
 D PRINT("ORCDSS") ; PRINT ORCDSS ORDERS  
 D PRINT("NON-ORCDSS") ; PRINT NON-ORCDSS ORDERS  
 K ^TMP($J)  
 D ^%ZISC ; CLOSE OUTPUT DEVICE  
 Q  
DIVMATCH(Z) ; DETERMINE IF ORDER IS FOR REQUESTED DIVISION  
 N ORLLOC,ORRQDIV  
 I ORDIV="" Q 1 ; NO DIVISIONS REQUESTED  
 S ORLLOC=$P(^OR(100,ORRD,0),U,10) I +ORLLOC>0,ORLLOC["SC" S ORRQDIV=$P(^SC(+ORLLOC,0),U,15)  
 I $G(ORRQDIV)>0,$D(^OR(100.953,ORDIV,1,"B",ORRQDIV)) Q 1  
 E  Q 0  
ANTIMIC(Z) ; DETERMINE IF ORDER HAS ANY ANTIMICROBIAL ITEMS  
 N ORI,ORJ,ORAM  
 S ORAM=0 ; INITIALIZE TO NON ANTI-MICROBIAL ORDER  
 S ORI=0 F  S ORI=$O(^OR(100,ORRD,.1,"B",ORI)) Q:+ORI'>0 I +$G(^ORD(101.43,ORI,618001))=1 S ORAM=1 ; ORDERABLE ITEM MARKED AS ANTI-MICROBIAL  
 Q ORAM  
ORCDSS(Z) ; DETERMINE IF ORDER IS ORCDSS OR NOT  
 ;LOOK FOR AUDIT IFN IN COMMENTS, STORE IFN OF ORDER IN AUDIT FILE  
 N ORK,ORKK,ORKK0,ORCDSSORDER  
 S ORCDSSORDER=0 ; INITIALIZE TO NON-ORCDSS ORDER  
 I ORKK0]"",ORKK0["\*\* Pharmacy Confirmation #: " S ORCDSSORDER=1\_"^"\_ORKK0 ;IDENTIFIED AS ORCDSS ORDER  
 I $D(^OR(100,ORJ,8,0)) S ORK=0 F  S ORK=$O(^OR(100,ORJ,8,ORK)) Q:+ORK'>0 I $D(^OR(100,ORJ,8,ORK,.1,0)) S ORKK=0 F  S ORKK=$O(^OR(100,ORJ,8,ORK,.1,ORKK)) Q:ORKK'>0 D  
 .S ORKK0=^OR(100,ORJ,8,ORK,.1,ORKK,0) I ORKK0["\*\* Pharmacy Confirmation" S ORCDSSORDER=1\_"^"\_ORKK0 ; IDENTIFIED AS ORCDSS ORDER  
 Q ORCDSSORDER  
PRINT(Z) ; PRINT RESULTS  
 ; ORWHO = 'WHO ENTERED' field  
 ; ORCNUM = Pharmacy Confirmation No (Record # in file 6189050)  
 ; OR2EDAT = 'WHEN ENTERED' field  
 ; OR2PAT = Patient Name and last 4 of SSN  
 ; OR2PRB = 'CURRENT AGENT/PROVIDER' field  
 ; OR2LOC = 'PATIENT LOCATION' field  
 N ORX0,X8,ORI,ORK,OR2DOTS,OR2PROV,ORCNUM,OR2EDAT,OR2HFS,OR2PAT,DFN,OR2SSN,OR2LOC,OR2OITEM  
 S OR2HFS=$$HFS() ; DETERMINE WHETHER PRINTER OR HFS DEVICE  
 I OR2HFS,Z="ORCDSS" W "Type"\_U\_"Order #"\_U\_"Patient"\_U\_"Who Entered"\_U\_"Confirm #"\_U\_"When Entered"\_U\_"Provider"\_U\_"Location"\_U\_"Orderable Item"  
 I 'OR2HFS W !!,"Anti-Microbial Orders",?65 D ^%D W:ORDIV]"" !,"Division Group: ",$P(^OR(100.953,ORDIV,0),"^",1) W !," ",ORRD," Orders" S $P(OR2DOTS,"-",81)="" D  
 .W !!,"Order #",?12,"Patient" W:ORZ="ORCDSS" ?30,"Confirm #" W ?42,"Who Entered",?68,"When Entered",!?12,"Location",?42,"Provider",!?42,"Orderable Item",!,OR2DOTS  
 S ORI=0 F  S ORI=$O(^TMP($J,ORRD,ORI)) Q:+ORI'>0 D  
 .S ORX0=^OR(100,ORI,0)  
 .S ORWHO=$P(ORX0,"^",6),ORWHO=$E($P(^VA(200,ORWHO,0),U,1),1,20)  
 .S ORCNUM=^TMP($J,ORRD,ORI) I ORCNUM]"" S ORCNUM=$P($P(ORCNUM,"Confirmation #: ",2)," ",1)  
 .;S ORCNUM=^TMP($J,ORRD,ORI) I ORCNUM]"" S ORCNUM=$P($P(ORCNUM,"Confirmation",2)," ",1)  
 .S OR2EDAT=$P(ORX0,U,7) I OR2EDAT<ORSDATE!(OR2EDAT>OREDATE) Q  ; ORDER NOT ENTERED WITHIN DATE RANGE  
 .S OR2PAT=$P(ORX0,U,2),DFN=+OR2PAT  
 .I +DFN>0 S OR2PAT=$P(^DPT(DFN,0),U,1),OR2SSN=$E($P(^(0),U,9),6,9),OR2PAT=$P(OR2PAT,",",1),OR2PAT=OR2PAT\_" ("\_OR2SSN\_")"  
 .E  S OR2PAT="No Patient"  
 .S OR2EDAT=$$FMTE^XLFDT(OR2EDAT,"2D")  
 .S OR2PROV=$P(ORX0,U,4) S:+OR2PROV>0 OR2PROV=$P(^VA(200,OR2PROV,0),U,1)  
 .S OR2LOC=$P(ORX0,U,10) I +OR2LOC>0 S X2=$P(OR2LOC,";",2) S OR2LOC=$P(@("^"\_X2\_+OR2LOC\_",0)"),U,1) .D ITEM  
 .I OR2HFS W !,ORZ\_U\_ORI\_U\_OR2PAT\_U\_ORWHO\_U\_ORCNUM\_U\_OR2EDAT\_U\_OR2PROV\_U\_OR2LOC\_U\_OR2OITEM  
 .E  W !,ORI,?12,OR2PAT,?32,ORCNUM,?42,ORWHO,?71,OR2EDAT,!?12,OR2LOC,?42,OR2PROV,!?42,OR2OITEM  
 Q  
HFS() ; RETURN '1' IF OUTPUTTING TO A FILE, '0' FOR DEVICE  
 I $P(^%ZIS(1,IOS,"TYPE"),U,1)="HFS" Q 1  
 Q 0  
ITEM   ;Define orderable item.  
 S OR2OITEM="",ORQQ=0  
 I $D(^OR(100,ORI,.1)) S ORI1=0 F  S ORI1=$O(^OR(100,ORI,.1,ORI1)) Q:ORI1'>0 Q:ORQQ  S OR2=^OR(100,ORI,.1,ORI1,0) D  
 .I $G(^ORD(101.43,OR2,618001))=1 S OR2OITEM=$P(^ORD(101.43,OR2,0),U,1),ORQQ=1  
 K ORQQ Q

**ORQOAUIC (New)**

ORQOAUIC ;ALB/RTW - QUICK ORDER DATA RETRIEVAL ; 11/24/16 1:54pm  
 ;;1.1;ORQOA QUICK ORDER AUDIT V;;Jun 08, 2015  
 Q  
AUD(ORDFN,TYPE) ; ENTRY POINT  
 N ORORN,ORPROV,ORLOC,ORDICDR,ORPROVDUZ,ORX,Y,ORD0,ORD1,ORD2,%,DIC,ORX0  
 Q:+ORDFN'>0 ""  ; Quit if no patient is specified per ticket CR643330  
 S ORORN=$G(^TMP("ZZ QUICK ORDER AUDIT",$J,"DLGID")) ; DEFINE QUICK ORDER IEN  
 S ORPROV=$G(^TMP("ZZ QUICK ORDER AUDIT",$J,"REC")) S ORLOC=$P(ORPROV,U,1),ORPROV=$P(ORPROV,U,4) ; DEFINE LOCATION AND   
  
PROVIDER  
 S ORPROVDUZ=$$PROVDUZ(ORPROV) ; GET PROVIDER'S DUZ  
 S ORX=$$NOW^XLFDT S DIC("DR")="1////"\_DUZ\_";2////"\_ORDFN\_";3////"\_ORORN\_";4////"\_ORPROVDUZ\_";5////"\_TYPE\_";6///"\_$G(ORLOC)  
 S ORDICDR=DIC("DR") ; WLS/MPLS NEW LINE 10/3/07  
 S DIC="^OR(100.95,",DIC(0)="L" D FILE^DICN  
 I +Y=-1 S ^WLS("ANAZQUIC",$H)=$G(ORDICDR) ; WLS/MPLS NEW LINE 10/3/07  
 Q "\*\* Pharmacy Confirmation #: "\_+Y\_" \*\*"  
 ; QUICK ORDER NUMBER  
 N ORQOORNAME,ORI,ORIFN  
 S ORIFN="<NONE>",ORQOORNAME=$P($G(ORX0),"^",1),ORQOORNAME=$E(ORQOORNAME,1,30)  
 S ORI=0 F  S ORI=$O(^ORD(101.41,"B",ORQOORNAME,ORI)) Q:+ORI'>0 I ^ORD(101.41,ORI,0)=ORX0 S ORIFN=I  
 Q ORIFN  
DRUG(ORD0) ; FOR 'DRUG' COMPUTED FIELD IN THE QUICK ORDER AUDIT FILE  
 N ORD1,ORD2  
 S ORD1=$O(^ORD(101.41,ORD0,6,"D",4,0))  
 I ORD1'>0 Q "<NONE>"  
 S ORD2=+^ORD(101.41,ORD0,6,ORD1,1)  
 Q $$OUTPUT^ORQOAUIC(ORD2)  
OUTPUT(ORY) ; -- Output Xform for Value field of Response multiple of Order Dialog file, 101.41  
 N ORDIALOG,ORP,ORZ S ORZ=ORY  
 S ORP=$P($G(^ORD(101.41,ORD0,6,ORD1,0)),U,2)  
 I ORP S ORDIALOG(ORP,0)=$P($G(^ORD(101.41,ORP,1)),U,1,2),ORDIALOG(ORP,1)=ORY,ORZ=$$EXT^ORCD(ORP,1)  
 Q ORZ  
PROVDUZ(ORUN) ; RETURN DUZ FOR USER ORUN WHERE ORUN IS USER'S ORNAME  
 ; If more than one user with this ORNAMe, pick one with access code  
 ; If more than one with access code, just pick first one  
 ; ORCNT = # of users with this ORNAMe  
 ; ORORACCNT = # of user with this ORNAMe that have an access code  
 N ORNAME,ORCNT,ORI,USER,ORACCNT  
 S ORNAME=$E(ORUN,1,35),(ORCNT,ORI,ORACCNT)=0 F  S ORI=$O(^VA(200,"B",ORNAME,ORI)) Q:+ORI'>0 S ORCNT=ORCNT+1 I $P($G(^VA(200,ORI,0)),U,3) S ORACCNT=ORACCNT+1 S ORACCNT(ORI)="" ; COUNT USERS WITH THIS ORNAME  
 I ORACCNT>0 Q $O(ORACCNT(0)) ; At least one with access code, return first one with DUZ  
 Q $O(^VA(200,"B",ORNAME,0)) ; No one with access code. Just return first user with this ORNAMe.