Existing Product Intake Program (EPIP)

Patch OR\*3.0\*431

Remediation Plan



Department of Veterans Affairs

**November 2016**

Version 2.0

Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 11/15/2016 | 2.0 | Updated entire document | EPIP Project Team |
| 08/19/2016 | 1.0 | Initial (draft) version | EPIP Project Team |

Table of Contents

[1. Introduction 1](#_Toc467158236)

[2. Purpose 1](#_Toc467158237)

[3. Patch Description 1](#_Toc467158238)

[3.1. Needs and Requirements 3](#_Toc467158239)

[4. Points of Contact 3](#_Toc467158240)

[5. Code Remediation 3](#_Toc467158241)

[5.1. Standards and Conventions 3](#_Toc467158242)

[5.2. Review and Analysis 4](#_Toc467158243)

[5.3. Coding Changes 4](#_Toc467158244)

[6. Testing 4](#_Toc467158245)

[6.1. Test Plan 4](#_Toc467158246)

[6.2. Test Environment 5](#_Toc467158247)

[6.3. Test Readiness Review 5](#_Toc467158248)

[6.4. Testing Phases 5](#_Toc467158249)

[6.4.1. Unit Testing 5](#_Toc467158250)

[6.4.2. Component Integration and Systems Testing (CI/ST) 5](#_Toc467158251)

[6.4.3. Functional Testing 5](#_Toc467158252)

[6.4.4. Regression Testing 5](#_Toc467158253)

[6.4.5. VA Section 508 Compliance Testing 6](#_Toc467158254)

[7. Documentation Remediation 6](#_Toc467158255)

[7.1. User Guides 6](#_Toc467158256)

[7.2. Installation Guides 6](#_Toc467158257)

[7.3. Technical Manuals 6](#_Toc467158258)

[7.4. Operations Manuals 6](#_Toc467158259)

[8. Project Reporting 6](#_Toc467158260)

[9. Project Schedule 7](#_Toc467158261)

[10. Deployment 7](#_Toc467158262)

[11. Sustainment Requirements 7](#_Toc467158263)

[12. Maintenance and Knowledge Transfer 7](#_Toc467158264)

[Appendix A: XINDEX Listing for MUMPS Code Changes 8](#_Toc467158265)

[Appendix B: Source Code Changes 10](#_Toc467158266)

# 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 and testing of the intake product code to be deployed as patch OR\*3.0\*431. This patch addresses the following NSRs:

* NSR20080317 *Default Encounter Location*

This NSR has been implemented locally at the VA Medical Centers in Boise VA, Hines IL, Richmond VA., and Seattle WA.

* NSR20150608 *Accession of Site-Supported Lab Test*

This NSR has been implemented locally at the following VA Medical Centers: VA Heartland - West (Kansas City, Columbia, Topeka, Leavenworth, Wichita); VA Heartland - East (St. Louis, Poplar Bluff, Marion); VA Northern Indiana Health Care System (Marion, Fort Wayne).

* NSR20141210 *CPRS Pop-Up Box*

This NSR has been implemented locally at the VA Medical Center in Baltimore MD.

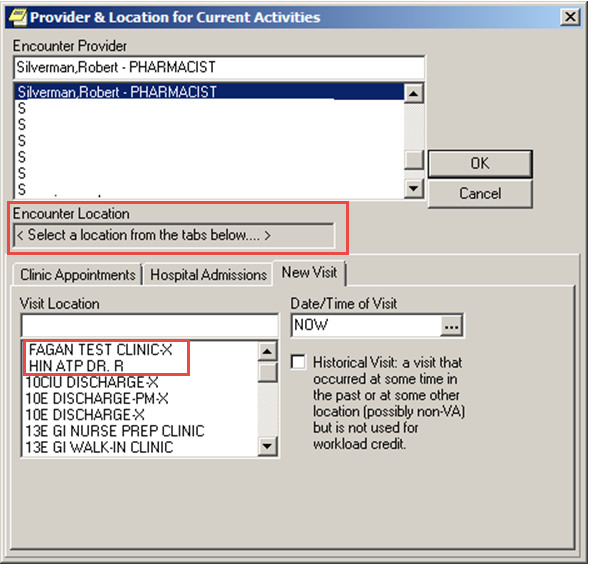
This document addresses the schedule, code remediation, testing, documentation, and delivery of this remediation effort.

# Patch Description

OR\*3.0\*431 provides the following enhancements to VistA:

* Enables healthcare providers to designate one or more default encounter locations to be displayed at the top of the provider’s Encounter Location selection list in the Computerized Patient Record System (CPRS) **Provider & Location for Current Activities** dialog box. Currently, an alphabetical list of all locations available in the HOSPITAL LOCATION file is displayed. This modification reduces the time necessary to select frequently used encounter locations and reduces the potential for erroneous selection of a location from a long list.

The modification also enables Clinical Application Coordinators (CACs) to select and manage default encounter locations on behalf of providers.



**Default Encounter Locations Displayed First**

* Restricts the display of available lab tests in CPRS to only those for which the ordering provider’s location matches the accession location for the test. Currently, all lab tests are allowed to be orderable items, even if the provider’s location does not match the accession location. If the provider inadvertently orders lab tests that cannot be performed at the local facility, then the tests are not accessioned, labels are not printed, and labs are not collected. There is no notification to the provider that an error has occurred.

This modification ensures that laboratory tests ordered at multi-divisional facilities can in fact be completed at the ordering provider’s local facility.

* Automatically displays a message box when a healthcare provider opens a patient chart in CPRS. This is a mechanism for communicating information that is not part of the patient’s official medical record. Messages can be global (the same message appearing for all patients), or specific to one or a select group of patients.

The messages are maintained through a series of file maintenance options, using the following system flags: FLAG 1/FLAG 2 (standard message to be sent to a select group of patients), STATE VETERANS HOME (indicates SVH residence for a patient), LOCAL NOTICE (specific message for a specific patient), COMBAT (indicates Operation Enduring Freedom (OEF)/Operation Iraqi Freedom (OIF) status), INELIGIBLE (indicates that the patient is ineligible for treatment at a VA facility), OBSERVATION (indicates that the patient is admitted and is in observation status), and NON-VESTED (indicates that the patient is eligible for a vesting visit).

## Needs and Requirements

The Needs and Requirements for the NSRs addressed in this remediation are:

NSR20080317 *Default Encounter Location*:

* NEED 385977: Encounters Location List Reduce typing and scrolling to select
* REQUIREMENT 392969: CPRS Location List Display preferred locations
* NEED 385979: Encounters Location List Reduce potential for selection error
* REQUIREMENT 392967: CPRS Location List Avoid errors in selecting from long lists
* NEED 725510: Designate Default Hospital Location

NSR20150608 *Accession of Site Supported Lab Test*:

* NEED 590351: Prevent Incorrect Ordering of Lab Tests
* NEED 576755: Facility Specific Lab Test
* NEED 590353: Prevent Physician Ordering Labs Incorrectly

NSR20141210 *CPRS Pop-Up Box*:

* NEED 508430: For authorized users of VA’s electronic health record who need to view patient-related information that is not readily available by simply viewing the patient chart. The ability to display a pop-up note upon selection of a patient that contains information.

# Points of Contact

The VA Point of Contact (POC) for NSR20080317 *Default Encounter Location* and NSR20141210 *CPRS Pop-Up Box* is Robert Silverman (PII),   
708-202-5040.

The DANPSC f r NSR20150608 *Accession of Site Supported Lab Test* is Liesl T Wilson, (PII), 816-861-4700.

# 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 http://URL 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 within CPRS, and 2) verification that the source code changes do not adversely affect any other VistA 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 NSR20080317 *Default Encounter Location* are in the following MUMPS routines:

**Modified routines:** ORWU, ORWU1

**New routines:** ORCLOC, ORCP031

The coding changes required for NSR20150608 *Accession of Site Supported Lab Test* are in the following MUMPS routines:

**Modified routines:** ORWDX

**New routines:** None

The coding changes required for NSR20141210 *CPRS Pop-Up Box* are in the following MUMPS routines:

**Modified routines:** ORWPT

**New routines:** ORPO7GUI, ORPOCHF, ORPOMDRO, ORPOOBS, ORPOTIO, ORPOVST

A detailed analysis of the coding changes 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 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 modification, the following tools will be leveraged: RQM, Reflections emulator, CPRS GUI v30b (1.0.30.72), 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.

### VA 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 http://www.DNS. 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 Guide: GUI Version*

## 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*

## 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 four 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.

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.

                   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    Oct 18, 2016@08:04:01

All Routines? No => No

Routine:

0 routines

Select BUILD NAME: OR\*3.0\*431       ORDER ENTRY/RESULTS REPORTING

Include the compiled template routines: N//

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

Save parameters in ROUTINE file? NO//

Index all called routines? NO//

DEVICE: ;;999  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    Oct 18, 2016@08:04:01

The BUILD file Data Dictionaries are being processed.

100.007 ORPU POPUP XECUTEABLE CODE..

100.00701 DESCRIPTION..

100.00702 XECUTABLE CODE..

100.00703 TEXT..

100.0071 ORPU POPUP PATIENT FLAG.

100.00711 FLAG.

100.00712 COMMENTS...

100.0072 ORPU POPUP FLAG..

100.0073 ORPU POPUP LOCAL NOTICE..

100.0074 ORPU POPUP VESTING.

200.08 DEFAULT ENCOUNTER LOCATION

The option and function files are being processed.

Routines are being processed.

Routines: 12  Faux Routines: 12

ORCLOC    ORCP031   ORPO7GUI  ORPOCHF   ORPOMDRO  ORPOOBS   ORPOTIO   ORPOVST

ORWDX     ORWPT     ORWU      ORWU1

          Data Dictionaries

|dd100.007     |dd100.00701   |dd100.00702   |dd100.00703   |dd100.0071

|dd100.00711   |dd100.00712   |dd100.0072    |dd100.0073    |dd100.0074

|dd200.08      |opt

--- CROSS REFERENCING ---

Compiled list of Errors and Warnings              Oct 18, 2016@08:04:01 page 1

No errors or warnings to report

--- END ---

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:** ORWU, ORWU1, ORWDX, ORWPT

**New routines:** ORCLOC, ORCP031, ORPO7GUI, ORPOCHF, ORPOMDRO, ORPOOBS, ORPOTIO, ORPOVST

**ORWU**

**Before:**

HOSPLOC(Y,FROM,DIR) ; Return a set of locations from HOSPITAL LOCATION

; .Y=returned list, FROM=text to $O from, DIR=$O direction,

N I,IEN,CNT S I=0,CNT=44

F Q:I'<CNT S FROM=$O(^SC("B",FROM),DIR) Q:FROM="" D ; IA# 10040.

. S IEN="" F S IEN=$O(^SC("B",FROM,IEN),DIR) Q:'IEN D

. . Q:("CW"'[$P($G(^SC(IEN,0)),U,3)!('$$ACTLOC(IEN)))

. . S I=I+1,Y(I)=IEN\_"^"\_FROM

Q

**After:**

HOSPLOC(Y,FROM,DIR) ; Return a set of locations from HOSPITAL LOCATION

; .Y=returned list, FROM=text to $O from, DIR=$O direction,

N I,IEN,CNT S I=0,CNT=44

I $D(^VA(200,DUZ,"DELOC")) D NEWLOC^ORCLOC(.Y,ORFROM,DIR) Q

F Q:I'<CNT S FROM=$O(^SC("B",FROM),DIR) Q:FROM="" D ; IA# 10040.

. S IEN="" F S IEN=$O(^SC("B",FROM,IEN),DIR) Q:'IEN D

. . Q:("CW"'[$P($G(^SC(IEN,0)),U,3)!('$$ACTLOC(IEN)))

. . S I=I+1,Y(I)=IEN\_"^"\_FROM

Q

**ORWU1**

**Before:**

NEWLOC(Y,ORFROM,DIR) ; Return "CZ" locations from HOSPITAL LOCATION file.

; C=Clinics, Z=Other, screened by $$ACTLOC^ORWU.

; .Y=returned list, ORFROM=text to $O from, DIR=$O direction.

N I,IEN,CNT S I=0,CNT=44

F Q:I'<CNT S ORFROM=$O(^SC("B",ORFROM),DIR) Q:ORFROM="" D ; IA# 100

40.

. S IEN="" F S IEN=$O(^SC("B",ORFROM,IEN),DIR) Q:'IEN D

. . Q:("C"'[$P($G(^SC(IEN,0)),U,3)!('$$ACTLOC^ORWU(IEN)))

. . S I=I+1,Y(I)=IEN\_"^"\_ORFROM

Q

**After:**

NEWLOC(Y,ORFROM,DIR) ; Return "CZ" locations from HOSPITAL LOCATION file.

; C=Clinics, Z=Other, screened by $$ACTLOC^ORWU.

; .Y=returned list, ORFROM=text to $O from, DIR=$O direction.

;;---------------------------------------------------------------

I $D(^VA(200,DUZ,"DELOC")) D NEWLOC^ORCLOC(.Y,ORFROM,DIR) Q

N I,IEN,CNT S I=0,CNT=44

F Q:I'<CNT S ORFROM=$O(^SC("B",ORFROM),DIR) Q:ORFROM="" D ; IA# 100

40.

. S IEN="" F S IEN=$O(^SC("B",ORFROM,IEN),DIR) Q:'IEN D

. . Q:("C"'[$P($G(^SC(IEN,0)),U,3)!('$$ACTLOC^ORWU(IEN)))

. . S I=I+1,Y(I)=IEN\_"^"\_ORFROM

Q

;

**ORWDX**

**Before:**

ORDITM(Y,FROM,DIR,XREF,QOCALL) ; Subset of orderable items

         ; Y(n)=IEN^.01 Name^.01 Name  -or-  IEN^Synonym <.01 Name>^.01 Name

         N I,IEN,CNT,X,DTXT,CURTM,DEFROUTE

         S DEFROUTE=""

         S QOCALL=+$G(QOCALL)

         S I=0,CNT=44,CURTM=$$NOW^XLFDT

         F  Q:I'<CNT  S FROM=$O(^ORD(101.43,XREF,FROM),DIR) Q:FROM=""  D

         . S IEN="" F  S IEN=$O(^ORD(101.43,XREF,FROM,IEN),DIR) Q:'IEN  D

         . . S X=^ORD(101.43,XREF,FROM,IEN)

         . . I +$P(X,U,3),$P(X,U,3)<CURTM Q

         . . I 'QOCALL,$P(X,U,5) Q

         . . S I=I+1

         . . I 'X S Y(I)=IEN\_U\_$P(X,U,2)\_U\_$P(X,U,2)

         . . E  S Y(I)=IEN\_U\_$P(X,U,2)\_$C(9)\_"<"\_$P(X,U,4)\_">"\_U\_$P(X,U,4)

         Q

ODITMBC(Y,XREF,ODLST) ;

**After:**

ORDITM(Y,FROM,DIR,XREF,QOCALL) ; Subset of orderable items

         ; Y(n)=IEN^.01 Name^.01 Name  -or-  IEN^Synonym <.01 Name>^.01 Name

         N I,IEN,CNT,X,DTXT,CURTM,DEFROUTE

         S DEFROUTE=""

         S QOCALL=+$G(QOCALL)

        S I=0,CNT=44,CURTM=$$NOW^XLFDT

         F  Q:I'<CNT  S FROM=$O(^ORD(101.43,XREF,FROM),DIR) Q:FROM=""  D

         . S IEN="" F  S IEN=$O(^ORD(101.43,XREF,FROM,IEN),DIR) Q:'IEN  D

         . . S X=^ORD(101.43,XREF,FROM,IEN)

         . . I +$P(X,U,3),$P(X,U,3)<CURTM Q

         . . I '$$START(XREF,IEN) Q

         . . I 'QOCALL,$P(X,U,5) Q

         . . S I=I+1

         . . I 'X S Y(I)=IEN\_U\_$P(X,U,2)\_U\_$P(X,U,2)

         . . E  S Y(I)=IEN\_U\_$P(X,U,2)\_$C(9)\_"<"\_$P(X,U,4)\_">"\_U\_$P(X,U,4)

         Q

         ;

START(INDEX,ET) ; Check to see if test is part of users DUZ(2)

         ;

         S OUT=1

         I INDEX="S.LAB" D

         . N NOD,P

         . S NOD=^ORD(101.43,ET,0),P=$P($P(NOD,U,2),";")

         . I '$D(^LAB(60,P)) Q

         . I '$D(^LAB(60,P,8)) Q

         . I '$D(^LAB(60,P,8,DUZ(2))) S OUT=0

         Q OUT

         ;

ODITMBC(Y,XREF,ODLST) ;

**ORWPT**

**Before:**

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

**After:**

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

D HXDATA^ORPO7GUI(.ORLST,DFN)

I $O(ORLST(0)) S ORLST(0)=1

Q

**ORCLOC (New)**

ORCLOC ;SLC/GRE - General Utilities for Windows Calls ; 22 Sep 2016 1:22 PM

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;Aug 7, 2002;Build 5

Q

;

NEWLOC(Y,ORFROM,DIR,ORCTYP) ; Return "CZ" locations from HOSPITAL LOCATION file.

; C=Clinics, W=Wards, Z=Other, screened by $$ACTLOC^ORWU.

; .Y=returned list, ORFROM=text to $O from, DIR=$O direction.

N %Y,ORC,ORCI,ORCIEN,ORCDUP S ORCI=0

D ; ONCE FOR PERSONAL LIST

.Q:ORFROM'=""

.N ORCIEN,ORCCNT S ORCCNT=44

. S ORC=0 F S ORC=$O(^VA(200,DUZ,"DELOC",ORC)) Q:'+ORC!(ORC'<ORCCNT)

D

. . S ORCIEN=$P($G(^VA(200,DUZ,"DELOC",ORC,0)),"^",1) Q:'ORCIEN

. . Q:("CWZ"'[$$GET1^DIQ(44,ORCIEN,2,"I")!('$$ACTLOC^ORWU(ORCIEN)))

. . S ORCI=ORCI+1,Y(ORCI)=ORCIEN\_"^ "\_$$GET1^DIQ(44,ORCIEN,.01)

. . S ORCDUP(ORCIEN)=""

D ; DAY-OF-WEEK CLINIC

.Q:ORFROM'=""

. N ORCENT,ORCPAR,X,ORCDOW,ORCDOWC

. S ORCENT="USR.`"\_DUZ

. S X=DT D DW^%DTC S ORCDOW=X

. S ORCPAR="ORLP DEFAULT CLINIC "\_ORCDOW

. S ORCDOWC=$$GET^XPAR(ORCENT,ORCPAR)

. I +ORCDOWC D ;

.. Q:("CWZ"'[$$GET1^DIQ(44,ORCDOWC,2,"I")!('$$ACTLOC^ORWU(ORCDOWC)))

.. Q:$D(ORCDUP(ORCDOWC))

.. S ORCDUP(ORCDOWC)=""

.. S ORCI=ORCI+1,Y(ORCI)=ORCDOWC\_"^ "\_$$GET1^DIQ(44,ORCDOWC,.01)

D ;TIU PREFERENCES DEFAULT LOCATION

. Q:ORFROM'=""

. N ORCTIU1,ORCTIU2

. Q:'$D(^TIU(8926,"B",DUZ))

. S ORCTIU1=$O(^TIU(8926,"B",DUZ,0)) Q:'+ORCTIU1

. S ORCTIU2=$$GET1^DIQ(8926,ORCTIU1,.02,"I") Q:'+ORCTIU2

. Q:("CWZ"'[$$GET1^DIQ(44,ORCTIU2,2,"I")!('$$ACTLOC^ORWU(ORCTIU2)))

. Q:$D(ORCDUP(ORCTIU2))

. S ORCDUP(ORCTIU2)=""

. S ORCI=ORCI+1,Y(ORCI)=ORCTIU2\_"^ "\_$$GET1^DIQ(44,ORCTIU2,.01)

D ;TIU DAY OF WEEK LOCATION

. Q:ORFROM'=""

. N ORCTIU1,ORCTIU2,ORCTIU3,ORCDOW,X

. S X=DT D H^%DTC S ORCDOW=%Y+1

. Q:'$D(^TIU(8926,"B",DUZ))

. S ORCTIU1=$O(^TIU(8926,"B",DUZ,0)) Q:'+ORCTIU1

. Q:'$D(^TIU(8926,ORCTIU1,1,"B",ORCDOW))

. S ORCTIU2=$O(^TIU(8926,ORCTIU1,1,"B",ORCDOW,0)) Q:'+ORCTIU2

. S ORCTIU3=$P(^TIU(8926,ORCTIU1,1,ORCTIU2,0),"^",2) Q:'+ORCTIU3

. Q:("CWZ"'[$$GET1^DIQ(44,ORCTIU3,2,"I")!('$$ACTLOC^ORWU(ORCTIU3)))

. Q:$D(ORCDUP(ORCTIU3))

. S ORCDUP(ORCTIU3)=""

. S ORCI=ORCI+1,Y(ORCI)=ORCTIU3\_"^ "\_$$GET1^DIQ(44,ORCTIU3,.01)

D ;Re-sort into alphabetical order

. N ORCJ,ORCDFE,ORCHOLD,ORCDFEIEN,ORCDFENAME,ORCJ2

. S ORCJ=0 F S ORCJ=$O(Y(ORCJ)) Q:'+ORCJ D

.. S ORCDFE=$G(Y(ORCJ)),ORCDFEIEN=$P(ORCDFE,U),ORCDFENAME=$P(ORCDFE,U,2

)

.. S ORCHOLD(ORCDFENAME,ORCJ,ORCDFEIEN)=""

. S ORCJ2=0

. S ORCDFENAME="" F S ORCDFENAME=$O(ORCHOLD(ORCDFENAME)) Q:ORCDFENAME'

]"" D

.. S ORCJ=0 F S ORCJ=$O(ORCHOLD(ORCDFENAME,ORCJ)) Q:'+ORCJ D

... S ORCDFEIEN=0 F S ORCDFEIEN=$O(ORCHOLD(ORCDFENAME,ORCJ,ORCDFEIEN))

Q:'+ORCDFEIEN D

.... S ORCJ2=ORCJ2+1 S Y(ORCJ2)=ORCDFEIEN\_U\_ORCDFENAME

D ; SECOND TIME FOR REGULAR LIST

.I $G(ORCTYP)']"" S ORCTYP="C"

.N ORCIEN,ORCCNT S ORCCNT=44

.F Q:ORCI'<ORCCNT S ORFROM=$O(^SC("B",ORFROM),DIR) Q:ORFROM="" D

.. S ORCIEN="" F S ORCIEN=$O(^SC("B",ORFROM,ORCIEN),DIR) Q:'ORCIEN D

... Q:(ORCTYP'[$$GET1^DIQ(44,ORCIEN,2,"I")!('$$ACTLOC^ORWU(ORCIEN)))

... S ORCI=ORCI+1,Y(ORCI)=ORCIEN\_"^"\_ORFROM

Q

;

FILEDIC(ORCDIC,ORCDIC0,ORCDICA,ORCDICB) ; Basic shell for DIC lookups

N X,Y,DTOUT,DUOUT,DIC

S DIC=ORCDIC,DIC(0)=ORCDIC0 S:$G(ORCDICA)]"" DIC("A")=ORCDICA S:$G(ORCD

ICB)]"" DIC("B")=ORCDICB

D ^DIC K DIC

S:Y>0 ORCFILES=+Y

Q

;

PARAM N ORCDUZ S ORCDUZ=DUZ

P2 N DIC,DIE,DR,DA,ILOC,ORC,ORCNONE

W @IOF

W !,"Now setting preferences for default HOSPITAL LOCATIONS for:"

W !?5,"--> ",$$GET1^DIQ(200,ORCDUZ,.01)

W !,"Currently selected locations are:"

S ILOC=0 F S ILOC=$O(^VA(200,ORCDUZ,"DELOC",ILOC)) Q:'+ILOC D

. S ORCLOC=$P(^VA(200,ORCDUZ,"DELOC",ILOC,0),"^")

. W !?5,$$GET1^DIQ(44,ORCLOC,.01)

. S ORCNONE=1

I '$G(ORCNONE) W !?5,"None selected..."

W !

P3 W !

S DIC="^VA(200,ORCDUZ,""DELOC"","

S DIC(0)="AEMQL"

S (DIC(1),DA(1))=ORCDUZ

D ^DIC

Q:Y=-1

S DIE=DIC K DIC

S DA(1)=ORCDUZ

S DA=+Y

S DR=.01

D ^DIE

K DIE,DR,DA,Y

G P3

;

OTHER N ORCDUZ

N DIC S DIC=200,DIC(0)="AEMQ" D ^DIC K DIC Q:+Y<1 S ORCDUZ=+Y

D P2

W !! G OTHER

OTHQU Q

;

**ORCP031 (New)**

ORCP031 ;EPIP/WLC - Patch 31 Post-install; 12 Sep 2016 ; 15 Sep 2016 9:37 AM

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;Sep 12, 2016

;

Q

;

POST ; -- post installation for OR\*3.0\*431

D OPADD

Q

;

OPADD ; add OR PCE options to Menus in OPTION file #19

D BMES^XPDUTL("Adding OR PCE options to menus in OPTION file #19")

;

N ORCOPT,ERR

S ORCOPT=$$FIND1^DIC(19,,"AMX","OR PCE DEFAULT LOCATION")

I ORCOPT D

. N DA

. N FDA,IENS,X,Y

. S X=$O(^DIC(19,"B","ORPO MENU",0))

. I $D(^DIC(19,X,10,"B",ORCOPT)) Q

. S Y="?+1,"

. S IENS=X\_","

. N REC S REC=$P($G(^DIC(19,X,10,0)),U,3)+1

. S FDA(19.01,"+"\_REC\_","\_X\_",",.01)=ORCOPT

. S FDA(19.01,"+"\_REC\_","\_X\_",",2)="DL"

. D UPDATE^DIE("","FDA",,.ERR)

. I $D(ERR) D BMES^XPDUTL("Error in adding to ORPO MENU")

K ORCOPT ; Add entry for Clinical Coordinator

S ORCOPT=$$FIND1^DIC(19,,"AMX","OR PCE DEFAULT LOC ADMIN")

I ORCOPT D

. N DA

. N FDA,IENS,X,Y

. S X=$O(^DIC(19,"B","OR PARAM COORDINATOR MENU",0))

. I $D(^DIC(19,X,10,"B",ORCOPT)) Q

. S Y="?+1,"

. S IENS=X\_","

. N REC S REC=$P($G(^DIC(19,X,10,0)),U,3)+1

. S FDA(19.01,"+"\_REC\_","\_X\_",",.01)=ORCOPT

. S FDA(19.01,"+"\_REC\_","\_X\_",",2)="DL"

. D UPDATE^DIE("","FDA",,.ERR)

. I $D(ERR) D BMES^XPDUTL("Error in adding to OR PARAM COORDINATOR MENU")

Q

;

**ORPO7GUI (New)**

ORPO7GUI ;HINES/RMS, REGION 1/KLD/RMM - CPRS CHART FLAGGING FOR GUI ; 6-1-01; 1/27/12 3:40 PM

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;7/30/2012;Build 9

;IA 10076 XUSEC

;IA 2324

;CHANGE THE VALUE FOR THE ORWOR AUTO CLOSE PT MSG (SYSTEM) PARAMETER TO ADJUST THE LENGTH OF TIME THE WINDOW IS OPEN 0=INDEFINITE

;called from ORPOPAGU,WHICH IS CALLED FROM ORWPT

HXDATA(LST,DFN) ;ENTRY POINT FROM ORWPT

EN ;FORMER ENTRY POINT FROM A7RDPAGU

N ORPOTI,ORPOQUIT,ILST S ILST=0

F ORPOTI("I")=0:0 S ORPOTI("I")=$O(^OR(100.007,ORPOTI("I"))) Q:'ORPOTI("I") D

.Q:$$GET1^DIQ(100.007,ORPOTI("I"),1)'="YES" ;Active

.K ORPOQUIT ;S ILST=0

.F ORPOTI("II")=0:0 S ORPOTI("II")=$O(^OR(100.007,ORPOTI("I"),2,ORPOTI("II"))) Q:'ORPOTI("II")!($D(ORPOQUIT)) D

..X ^OR(100.007,ORPOTI("I"),2,ORPOTI("II"),0)

Q

;

INC S ILST=$G(ILST)+1

Q

;LST USED BY CPRS GUI SOFTWARE

NULL S LST(ILST)=" "

Q

FL(ORPODFN,ORPOFL) ;CHECK IF PATIENT HAS FLAG

;ORPODFN=PATIENT DFN

;ORPOFL=FLAG YOU ARE LOOKING FOR

N ORPOI,ORPOR S ORPOR=0

F ORPOI=0:0 S ORPOI=$O(^OR(100.0071,ORPODFN,1,ORPOI)) Q:'ORPOI D

.S:$$GET1^DIQ(100.00711,ORPOI\_","\_ORPODFN,.01)=ORPOFL ORPOR=1

Q ORPOR

ADDT(X1,X2) ;ADD/SUBTRACT FROM DATE

N X D C^%DTC

Q X

FDT(Y) ;FORMAT INTERNAL TO EXTERNAL DATE

D DD^%DT

Q Y

TXT ; PRINT TEXT

N ORPOI D INC,NULL

F ORPOI=0:0:3 S ORPOI=$O(^OR(100.007,ORPOTI("I"),3,ORPOI)) Q:'ORPOI D

.D INC S LST(ILST)=^OR(100.007,ORPOTI("I"),3,ORPOI,0)

.D:LST(ILST)["|" VAR(LST(ILST))

Q

VAR(ORPO) ;REMOVE ~ PRINT VARIABLE

N ORPOI,ORPOT,ORPOVAR

F ORPOI=0:0 S ORPOT=$F(ORPO,"|") Q:'ORPOT D

.S ORPOVAR=$P(ORPO,"|",2),ORPO=$P(ORPO,"|")\_@ORPOVAR\_$P(ORPO,"|",3,200)

S LST(ILST)=ORPO

Q

GFY(ORPODT) ; GET FISCAL YEAR

N ORPOMO,ORPOYR

S ORPOMO=$E(ORPODT,4,5),ORPOYR=$E(ORPODT,1,3)

S ORPOYR=$S(ORPOMO>9:ORPOYR+1,1:ORPOYR)

S ORPOYR=$S($E(ORPOYR)=2:19\_$E(ORPOYR,2,3),$E(ORPOYR)=3:20\_$E(ORPOYR,2,3),$E(ORPOYR)=4:21\_$E(ORPOYR,2,3),1:0000)

Q ORPOYR

FLAGOK(TYPE) ;RMS/HINES 3-3-04 TO CONTROL NUMBER OF FLAG VIEWS PER DAY

N ORPOFDAT,X,X1,X2

S X1=DT,X2=+1 D C^%DTC S ORPOFDAT=X

S ^XTMP("ORPOFLAG"\_DT,0)=ORPOFDAT\_U\_DT\_U\_"Pop-Up Flag Daily Usage Data"

Q $G(^XTMP("ORPOFLAG"\_DT,TYPE,DUZ,+$G(DFN)))

USER(ORPODUZ) Q:$$ISA^USRLM(ORPODUZ,"PHYSICIAN",.ORPOERR) 1

Q:$$ISA^USRLM(ORPODUZ,"PHYSICIAN ASSISTANT",.ORPOERR) 1

Q:$$ISA^USRLM(ORPODUZ,"NURSE PRACTITIONER",.ORPOERR) 1

Q:$$ISA^USRLM(ORPODUZ,"MEDICAL STUDENT",.ORPOERR) 1

Q:$D(^XUSEC("ORES",ORPODUZ)) 1

Q 0

;

**ORPOCHF (New)**

ORPOCHF ;R01/RMM - Pop-Up for Congestive heart failure in CPRS ;12/4/2013

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;;Build 9

;find patients discharged within 30 days with a primary diagnosis of CHF icd 9 code of 428.x

;when icd 10 is released this will have to be changed

;359 NAME: DBIA359

Q ;QUIT IF NOT ENTRY POINT

EN(ORPODFN) ;ENTRY POINT, PATIENT DFN

N ORPOI,ORPOSDT,ORPORET S ORPORET=0 K ^TMP("DILIST",$J)

S ORPOSDT=$$ADDT(DT,-30)

D FIND^DIC(45,,"@;.01I;79;70I","Q",ORPODFN,,"B") ;PTF FILE

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI D

.Q:^TMP("DILIST",$J,"ID",ORPOI,70)']""

.Q:^TMP("DILIST",$J,"ID",ORPOI,70)<ORPOSDT

.Q:^TMP("DILIST",$J,"ID",ORPOI,79)'["428."

.S ORPORET=1

Q ORPORET

ADDT(X1,X2) ;ADD/SUBTRACT FROM DATE

N X D C^%DTC

Q X

**ORPOMDRO (New)**

ORPOMDRO ;R01/RMM - POP-Up FOR MRSA/MDRO in CPRS ;4/8/2013

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;;Build 9

;could not find an ICA for file 104.1

;

EN(DFN) ;ENTRY POINT

N ORPOI,ORPOII,ORPOTEST,ORPO,ORPOA,ORPORET,ORPOIND,ORPORES,ORPOF,ORPOVAL,ORPOIII,ORPOD0 S ORPORET=0,ORPOF=0

D LIST^DIC(104.1,,"@;.01IE","Q",,,,"B") ;MRSA TOOLS LAB SEARCH/EXTRACT FILE

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI!(ORPORET=1) D

.S ORPOD0=^TMP("DILIST",$J,2,ORPOI)

.D LIST^DIC(104.15,","\_ORPOD0\_",","@;.01IE;1;2","Q",,,,"B",,,"ORPO")

.K ORPOA S ORPOF=0

.F ORPOII=0:0 S ORPOII=$O(ORPO("DILIST",2,ORPOII)) Q:'ORPOII!(ORPORET=1) D

..S ORPORET=$$FTEST(ORPO("DILIST","ID",ORPOII,.01,"I"))

..Q:ORPORET=1

..S ORPOIND=ORPO("DILIST","ID",ORPOII,1)

..S:ORPOIND="Contains" ORPOIND="[",ORPOF=1 S:ORPOIND="Greater Than" ORPOIND=">",ORPOF=1

..S:ORPOIND="Less Than" ORPOIND="<",ORPOF=1 S:ORPOIND="Equal To" ORPOIND="=",ORPOF=1

..Q:ORPOF=0

..S ORPOD1=ORPO("DILIST",2,ORPOII),ORPOTEST=ORPO("DILIST","ID",ORPOII,.01,"I")

..Q:$$GET1^DIQ(60,ORPOTEST,400)=""

..S ORPO=$$ONE^ORPOTIO($$GET1^DIQ(60,ORPOTEST,400)\_"^100^1Y")

..F ORPOIII=0:0 S ORPOIII=$O(^TMP("ORPOTIOB2",$J,ORPOIII)) Q:'ORPOIII D

...S ORPORES=""""\_$P(^TMP("ORPOTIOB2",$J,ORPOIII,0),"@",2)\_""""

...S ORPOVAL=""""\_ORPO("DILIST","ID",ORPOII,2)\_""""

...Q:ORPORES=""""""

...Q:ORPOVAL=""""""

...S ORPORES=$TR(ORPORES,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLNMOPQRSTUVWXYZ")

...S ORPOVAL=$TR(ORPOVAL,"abcdefghijklmnopqrstuvwxyz","ABCDEFGHIJKLNMOPQRSTUVWXYZ")

...I @(ORPORES\_ORPOIND\_ORPOVAL) S ORPORET=1 ;"\*\*\*\* MDRO PRECAUTIONS \*\*\*\*"

W !,"EN: ",ORPORET

Q ORPORET

FTEST(ORPOT) ; FIND MICROBIOLOGY TEST

N ORPOLRDFN,ORPOI,ORPOII,ORPOET,R2,R3,ORPOAS,ORPOD1,ORPORET S ORPORET=0

D LIST^DIC(104.109,","\_ORPOD0\_",","@;.01IE;","Q",,,,"B",,,"R2") ;ETIOLOGY MULTIPLE

F ORPOI=0:0 S ORPOI=$O(R2("DILIST",2,ORPOI)) Q:'ORPOI D

.K ORPOET S ORPOET=R2("DILIST","ID",ORPOI,.01,"I"),ORPOD1=R2("DILIST",2,ORPOI)

.D LIST^DIC(104.191,","\_ORPOD1\_","\_ORPOD0\_",","@;.01;1;2","Q",,,,"B",,,"R3") ;ANTIMICROBIAL SUSCEPTIBILITY MULTIPLE

.F ORPOII=0:0 S ORPOII=$O(R3("DILIST",2,ORPOII)) Q:'ORPOII!(ORPORET=1) D

..S ORPOET=R2("DILIST","ID",ORPOI,.01,"I")\_U\_R3("DILIST","ID",ORPOII,.01)\_U\_R3("DILIST","ID",ORPOII,2)

..S:R3("DILIST","ID",ORPOII,1)="Contains" ORPOET=ORPOET\_U\_"["

..S:R3("DILIST","ID",ORPOII,1)="Greater Than" ORPOET=ORPOET\_U\_">"

..S:R3("DILIST","ID",ORPOII,1)="Less Than" ORPOET=ORPOET\_U\_"<"

..S:R3("DILIST","ID",ORPOII,1)="Equal To" ORPOET=ORPOET\_U\_"="

..S ORPORET=$$GORG(ORPOET) Q:ORPORET=1

..S ORPORET=$$GMYC(ORPOET) Q:ORPORET=1

W !,"FORG: ",ORPORET

Q ORPORET

GORG(ORPOE) ;GET ORGANISM

N ORPOLRDFN,ORPOBDT,ORPOEDT,ORPOBRDT,ORPOERDT,ORPOI,ORPOD1,ORPOD2,ORPOD,ORPORET

S ORPOLRDFN=$$LRDFN^LRPXAPIU(DFN),ORPORET=0

S ORPOBDT=$$ADDT(DT,-365),ORPOEDT=DT

S ORPOBRDT=9999999-ORPOBDT,ORPOERDT=9999999-ORPOEDT

F ORPOD1=ORPOERDT:0:(ORPOBRDT\_.9999) S ORPOD1=$O(^LR(ORPOLRDFN,"MI",ORPOD1)) Q:'ORPOD1 D ;LAB DATA FILE MICROBIOLOGY MULTIPLE

.F ORPOD2=0:0 S ORPOD2=$O(^LR(ORPOLRDFN,"MI",ORPOD1,3,ORPOD2)) Q:'ORPOD2 D

..D:$P(ORPOE,U)=$P(^LR(ORPOLRDFN,"MI",ORPOD1,3,ORPOD2,0),U)

...S ORPOD=0,ORPOD=$O(^DD(63.3,"B",$P(ORPOE,U,2),ORPOD))

...I @(""""\_$P(ORPOE,U,3)\_""""\_$P(ORPOE,U,4)\_""""\_$$GET1^DIQ(63.3,ORPOD2\_","\_ORPOD1\_","\_ORPOLRDFN,ORPOD)\_"""") S ORPORET=1

W !,"GORG: ",ORPORET

Q ORPORET

GMYC(ORPOE) ;GET MYCOBACTERIUM ;^LR(D0,MI,D1,12,D2,0)= (#.01) MYCOBACTERIUM [1P:61.2] ^ (#1) QUANTITY [2F] ^

N ORPOLRDFN,ORPOBDT,ORPOEDT,ORPOBRDT,ORPOERDT,ORPOI,ORPOD1,ORPOD2,ORPOD,ORPORET

S ORPOLRDFN=$$LRDFN^LRPXAPIU(DFN),ORPORET=0

S ORPOBDT=$$ADDT(DT,-365),ORPOEDT=DT

S ORPOBRDT=9999999-ORPOBDT,ORPOERDT=9999999-ORPOEDT

F ORPOD1=ORPOERDT:0:(ORPOBRDT\_.9999) S ORPOD1=$O(^LR(ORPOLRDFN,"MI",ORPOD1)) Q:'ORPOD1 D ;LAB DATA FILE MICROBIOLOGY MULTIPLE

.F ORPOD2=0:0 S ORPOD2=$O(^LR(ORPOLRDFN,"MI",ORPOD1,12,ORPOD2)) Q:'ORPOD2 D

..D:$P(ORPOE,U)=$P(^LR(ORPOLRDFN,"MI",ORPOD1,12,ORPOD2,0),U)

...S ORPOD=0,ORPOD=$O(^DD(63.39,"B",$P(ORPOE,U,2),ORPOD))

...I @(""""\_$P(ORPOE,U,3)\_""""\_$P(ORPOE,U,4)\_""""\_$$GET1^DIQ(63.39,ORPOD2\_","\_ORPOD1\_","\_ORPOLRDFN,ORPOD)\_"""") S ORPORET=1

W !,"GMYC: ",ORPORET

Q ORPORET

ADDT(X1,X2) ;ADD/SUBTRACT FROM DATE

N X D C^%DTC

Q X

**ORPOOBS (New)**

ORPOOBS ;R01/HAM3,RMM - Pop-Up for OBSERVATION in CPRS ;07/30/2012

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;7/30/2012;Build 9

;

GETADMFM(DFN) ; GET THE FILEMAN FORMAT OF THE ADMISSION DATE

N VAIN

D INP^VADPT

Q +VAIN(7)

GMT(ORPOMX) ;GET MAX TIME FROM TEXT FIELD

N ORPOI

F ORPOI=0:0 S ORPOI=$O(^OR(100.007,ORPOTI("I"),3,ORPOI)) Q:'ORPOI D

.S:^OR(100.007,ORPOTI("I"),3,ORPOI,0)["MAX TIME" ORPOMX=$P(^OR(100.007,ORPOTI("I"),3,ORPOI,0),"=",2)

Q

GETDHSD(ORPOFMDT) ; GET THE DECIMAL TIME SINCE A FILEMAN DATE, ROUNDS SECONDS UP 15.331 = 15.34

Q $FN($$FMDIFF^XLFDT($$NOW^XLFDT,ORPOFMDT,2)/3600,"",2)

GETTMLFT(ORPODATE,ORPOMAXT) ; get the time left

N ORPOTIME

S ORPOTIME=$FN($$FMDIFF^XLFDT($$NOW^XLFDT,ORPODATE,2)/3600,"",2)

Q $$GETTXT3(ORPOTIME,ORPOMAXT)

GETTEXT(LST,DFN) ; GENERATE THE LST ARRAY TO BE USED BASED ON THE ADMIT DATE

N ORPOADT,ORPOMXT,ORPOOB,ORPOTM S ORPOMXT=0

D INC^ORPO7GUI,NULL^ORPO7GUI

;

;change for directive 1036

;S ORPOMXT=23+(59/60) ; MAX HOURS AND 59 MINUTES ; MAX ALLOWED TIME

D GMT(.ORPOMXT) ;GET MAX TIME FROM TEXT FIELD

S:ORPOMXT=0 ORPOMXT=48 ;IF MAX TIME NOT DEFINED IN TEXT FIELD SET TO 48 HRS

S ORPOMXT=(ORPOMXT-1)+(59/60) ; MAX HOURS AND 59 MINUTES ; MAX ALLOWED TIME

;

S ORPOADT=$$GETADMFM(DFN) ;GET ADMIT DATETIME

D GETTXT2(.LST,ORPOADT,ORPOMXT)

Q

GETTXT2(LST,ORPOADT,ORPOMXT) ;

N ORPOADTX,ORPOOB,ORPOTM

S ORPOADTX=$$FMTE^XLFDT(ORPOADT)

S ORPOOB=$$GETTMLFT(ORPOADT,ORPOMXT) ;GET TIME LEFT

S ORPOTM=$$GETDHSD(ORPOADT) ;get decimal time

I ORPOOB["EXCEEDED" D

.D INC^ORPO7GUI S LST(ILST)="DISCHARGE OR CHANGE OBSERVATION TO INPT STATUS NOW!"

.D INC^ORPO7GUI S LST(ILST)=ORPOOB

.D INC^ORPO7GUI S LST(ILST)="Observation admit was at: "\_ORPOADTX

E D

.;

.;change for directive 1036

.;I ORPOTM>=23 D

.I ORPOTM>=$P(ORPOMXT,".") D

..;

..;D INC^ORPO7GUI S LST(ILST)="23hr OBSERVATION PERIOD IS OVER!!"

..D INC^ORPO7GUI S LST(ILST)=$P(ORPOMXT,".")\_"th HOUR OF OBSERVATION IS OVER!"

..D INC^ORPO7GUI S LST(ILST)="DISCHARGE OR CHANGE OBSERVATION TO INPT STATUS NOW!"

..D INC^ORPO7GUI S LST(ILST)=ORPOOB

.E D

..D INC^ORPO7GUI S LST(ILST)="OBSERVATION ADMIT AT: "\_ORPOADTX

..;

..;change for directive 1036

..;I ORPOTM>=20 D

..I ORPOTM>=($P(ORPOMXT,".")-3) D

...;

...D INC^ORPO7GUI S LST(ILST)="MAKE PLANS FOR DISCHARGE OR FULL ADMIT."

...D INC^ORPO7GUI S LST(ILST)=ORPOOB

..E D

...I ORPOTM>0 D

....D INC^ORPO7GUI S LST(ILST)=ORPOOB

Q

GETTXT3(ORPODECTIME,ORPOMAXTIME) ;

N ORPODIFF,ORPOHRS,ORPOMINS,ORPORESULT

S ORPORESULT=""

S ORPODIFF=+$FN(ORPOMAXTIME-ORPODECTIME,"",2)

S ORPOHRS=+$P(ORPODIFF,".",1)

S ORPOMINS=$FN((ORPODIFF-ORPOHRS)\*60,"",0)

I ORPODIFF>0 S ORPORESULT="Discharge or admit within: "\_ORPOHRS\_" hour"\_$S(ORPOHRS=1:"",1:"s")\_" and "\_ORPOMINS\_" minute"\_$S(ORPOMI

NS=1:"",1:"s")

I ORPODIFF=0 S ORPORESULT="Discharge or admit within: "\_ORPOHRS\_" hour"\_$S(ORPOHRS=1:"",1:"s")\_" and "\_ORPOMINS\_" minute"\_$S(ORPOMI

NS=1:"",1:"s")

I ORPODIFF<0 S ORPORESULT="OBSERVATION EXCEEDED by: "\_-ORPOHRS\_" hour"\_$S(ORPOHRS=-1:"",1:"s")\_" and "\_-ORPOMINS\_" minute"\_$S(ORPOM

INS=-1:"",1:"s")

Q ORPORESULT

;

**ORPOTIO (New)**

ORPOTIO ; PHOENIX/KLD - Pop-Up for TIU OBJECTS - LAB TESTS & PANELS (TRENDS) in CPRS ; 5/25/12 3:13 PM

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;;Build 1

;;IAs used - 4245, 4246

ST Q

;

PANEL(X) ;Panel Lab Test in a time period object (time=nM, nD, or nY)

;X should be "Display name^# of occurances^time period^print a second line? (0 or 1)^Test IENS from file 63.04"

;Example: X="Chem 7^3^2Y^1^2,3,4:1:8,790"

N ORPOTI S ORPOTI("C")=0,$P(ORPOTI("SP")," ",30)=""

S ORPOTI("TN")=$P(X,U,1,2),ORPOTI("T")=$P(X,U,3),ORPOTI("LINE2")=$P(X,U,4),ORPOTI("TEST")=$P(X,U,5)

S ORPOTI("CHK",1)=$P(ORPOTI("TEST"),","),ORPOTI("CHK",2)=$P(ORPOTI("TEST"),",",2)

F ORPOTI("I")=1,2 S:ORPOTI("CHK",ORPOTI("I"))[":" ORPOTI("CHK",ORPOTI("I"))=$P(ORPOTI("CHK",ORPOTI("I")),":")

S:'ORPOTI("CHK",2)&(ORPOTI("CHK",1)) ORPOTI("CHK",2)=ORPOTI("CHK",1)

F ORPOTI("I")=1:1:$P(ORPOTI("TN"),U,2) S ORPOTI("TEST",ORPOTI("I"))=0 D

.X "F ORPOTI(""II"")="\_ORPOTI("TEST")\_" S ORPOTI(""TEST"",ORPOTI(""I""),ORPOTI(""II""))="""" S ORPOTI(""VALIDTESTS"",$$TEST^LRPXAPI

U(ORPOTI(""II"")))=ORPOTI(""II"")"

D GET I ORPOTI("TEST",1) D H(0),DAT(0),SET("") D

.I ORPOTI("LINE2") S ORPOTI("HOLD",1)=ORPOTI("HOLD") D H(ORPOTI("HOLD")),DAT(ORPOTI("HOLD",1))

Q "~@^TMP(""ORPOTIOB2"","\_$J\_")"

;

ONE(X) ;Single lab test in a time period object.

;X should be "Data name^# of occurances^time period (nM, nD, or nY)"

;or X could be "Print string^# of occurances^time period (nM, nD, or nY)^Data name number^Print completed time"

N ORPOTI S ORPOTI("TN")=X,ORPOTI("C")=0,$P(ORPOTI("SP")," ",50)=""

S ORPOTI("N")=$P(ORPOTI("TN"),U,2),ORPOTI("T")=$P(ORPOTI("TN"),U,3)

S:'ORPOTI("N") ORPOTI("N")=99 S:ORPOTI("T")="" ORPOTI("T")="99Y"

S:'$P(ORPOTI("TN"),U,4) ORPOTI("TEST")=$O(^DD(63.04,"B",$P(ORPOTI("TN"),U),0))

S:$P(ORPOTI("TN"),U,4) ORPOTI("TEST")=$P(ORPOTI("TN"),U,4)

I 'ORPOTI("TEST") D Q "~@^TMP(""ORPOTIOB2"","\_$J\_")"

.D K S ^TMP("ORPOTIOB2",$J,1,0)=$P(ORPOTI("TN"),U)\_" - INVALID TEST NAME"

F ORPOTI("I")=1:1:ORPOTI("N") S ORPOTI("TEST",ORPOTI("I"))=0,ORPOTI("TEST",ORPOTI("I"),ORPOTI("TEST"))=""

S X=$$TEST^LRPXAPIU(ORPOTI("TEST")),ORPOTI("VALIDTESTS",X)=ORPOTI("TEST"),ORPOTI("VALIDTESTS","B",ORPOTI("TEST"))=X ;IA 4246

S (ORPOTI("CHK",1),ORPOTI("CHK",2))=ORPOTI("TEST") D GET

D:$P(ORPOTI("TN"),U,5) ;also display Verify Date

.F ORPOTI("I")=9E9:0 S ORPOTI("I")=$O(^TMP("ORPOTIOB2",$J,ORPOTI("I")),-1) Q:'ORPOTI("I") D

..S ^TMP("ORPOTIOB2",$J,ORPOTI("I")+2,0)=^TMP("ORPOTIOB2",$J,ORPOTI("I"),0)

.S ^TMP("ORPOTIOB2",$J,1,0)=" TEST COLLECTION DATE RESULT VERIFY DATE"

.S ^TMP("ORPOTIOB2",$J,2,0)=""

ONEQ Q "~@^TMP(""ORPOTIOB2"","\_$J\_")"

;

GET ;Get data from ^LR(DFN,"CH")

N ORPOTITEST,LRDFN,T,X S T=ORPOTI("T") D K,NONE

S ORPOTI("N")=1

D RESULTS^LRPXAPI(.ORPOTITEST,DFN,"C",999,"","",DT,ORPOTI("ED")) ;IA 4245

F ORPOTI("I")=0:0 S ORPOTI("I")=$O(ORPOTI("VALIDTESTS",ORPOTI("I"))) Q:'ORPOTI("I") D

.S ORPOTI("VALIDTESTS","B",ORPOTI("VALIDTESTS",ORPOTI("I")))=ORPOTI("I")

S X="" F S X=$O(ORPOTITEST(X)) Q:X="" D

.Q:'$P(ORPOTITEST(X),U,2) Q:'$D(ORPOTI("VALIDTESTS",$P(ORPOTITEST(X),U,2)))

.S ^TMP("ORPOTIOB2",$J,"SORT",-ORPOTITEST(X),$P(ORPOTITEST(X),U,2))=$P(ORPOTITEST(X),U,4,5)

F ORPOTI("I")=-9E9:0 S ORPOTI("I")=$O(^TMP("ORPOTIOB2",$J,"SORT",ORPOTI("I"))) Q:'ORPOTI("I") D

.S ORPOTI("FLAG")=0

.F ORPOTI("II")=0:0 S ORPOTI("II")=$O(^TMP("ORPOTIOB2",$J,"SORT",ORPOTI("I"),ORPOTI("II"))) Q:'ORPOTI("II") D

..Q:'$D(^TMP("ORPOTIOB2",$J,"SORT",ORPOTI("I"),ORPOTI("VALIDTESTS","B",ORPOTI("CHK",1))))!('$D(^TMP("ORPOTIOB2",$J,"SORT",ORPOTI("I

"),ORPOTI("VALIDTESTS","B",ORPOTI("CHK",2)))))

..S ORPOTI("TEST")=ORPOTI("VALIDTESTS",ORPOTI("II")) Q:'$D(ORPOTI("TEST",ORPOTI("N"),ORPOTI("TEST")))

..S:'ORPOTI("TEST",ORPOTI("N"),ORPOTI("TEST")) ORPOTI("TEST",ORPOTI("N"),ORPOTI("TEST"))=^TMP("ORPOTIOB2",$J,"SORT",ORPOTI("I"),ORP

OTI("II")),ORPOTI("FLAG")=1

.S:ORPOTI("FLAG") ORPOTI("TEST",ORPOTI("N"))=-ORPOTI("I"),ORPOTI("N")=ORPOTI("N")+1

K ^TMP("ORPOTIOB2",$J,"SORT") Q

;

H(N) ;Header line

N X S X=$E($E($P(ORPOTI("TN"),U),1,11)\_" Coll. date"\_ORPOTI("SP"),1,23)

F ORPOTI("I")=N:0 S ORPOTI("I")=$O(ORPOTI("TEST",1,ORPOTI("I"))) Q:'ORPOTI("I")!($L(X)>72) D

.S ORPOTI("XX")=ORPOTI("SP")

.S:ORPOTI("XX")="" ORPOTI("XX")=$$LRDNM^LRPXAPIU(ORPOTI("I")),ORPOTI("XX")=$E($S(ORPOTI("XX")]"":ORPOTI("XX"),1:"Unknown"),1,8)\_ORP

OTI("SP") ;IA 4246

.S X=X\_$E(ORPOTI("XX"),1,7)\_" " Q:$L(X)>72

D SET(X) S ORPOTI("HOLD")=ORPOTI("I")-.1 Q

;

DAT(N) ;Data line

N X F ORPOTI("I")=1:1:$P(ORPOTI("TN"),U,2) Q:'ORPOTI("TEST",ORPOTI("I")) D D:$L(X)>72 SET(X)

.S X=$$CONV2(ORPOTI("TEST",ORPOTI("I")))\_ORPOTI("SP"),X=$E(X,1,23)

.F ORPOTI("TEST")=N:0 S ORPOTI("TEST")=$O(ORPOTI("TEST",ORPOTI("I"),ORPOTI("TEST"))) D:'ORPOTI("TEST")&($L(X)<73) SET(X) Q:'ORPOTI(

"TEST") D Q:$L(X)>72

..S ORPOTI("XX")=$P(ORPOTI("TEST",ORPOTI("I"),ORPOTI("TEST")),U) S:ORPOTI("XX")>0&(ORPOTI("XX")<1)&($E(ORPOTI("XX"))=".") ORPOTI("X

X")=0\_ORPOTI("XX")

..S:$P(ORPOTI("TEST",ORPOTI("I"),ORPOTI("TEST")),U,2)]"" ORPOTI("XX")=ORPOTI("XX")\_" "\_$P(ORPOTI("TEST",ORPOTI("I"),ORPOTI("TEST"))

,U,2)

..S:$E(ORPOTI("XX"),8)?1A ORPOTI("XX")=$E(ORPOTI("XX"),1,7)\_" " S X=X\_$E(ORPOTI("XX")\_ORPOTI("SP"),1,8)

Q

;

CONV() Q $$CONV2($$LRIDT^LRPXAPIU(ORPOTI("TEST",ORPOTI("I")))) ;IA 4246

CONV2(X) S ORPOTI("XX")=$E($P(X,".",2)\_"0000",1,4)

S X=X\_$E(ORPOTI("XX"),1,2)\_":"\_$E(ORPOTI("XX"),3,4)

S X=$E(X,4,5)\_"/"\_$E(X,6,7)\_"/"\_$E(X,2,3)\_" @ "

S X=X\_$E(ORPOTI("XX"),1,2)\_":"\_$E(ORPOTI("XX"),3,4) Q X

;

SET(X) S ORPOTI("C")=ORPOTI("C")+1,^TMP("ORPOTIOB2",$J,ORPOTI("C"),0)=X,X="" Q

;

AGO N X1,X2 S:'$D(ORPOTI("T")) ORPOTI("T")=T

S X1=DT,X2=+ORPOTI("T"),X=$P(ORPOTI("T"),X2,2),X2=-X2

S X2=X2\*$S(X="M":30,X="W":7,X="D":1,1:365)

D C^%DTC S ORPOTI("ED")=$$LRIDT^LRPXAPIU(X) Q ;IA 4246

;

K K ^TMP("ORPOTIOB2",$J) Q

NONE S ^TMP("ORPOTIOB2",$J,1,0)=$P(ORPOTI("TN"),U)\_" - NONE FOUND" Q

D(Y) D DD^%DT Q Y

**ORPOVST (New)**

ORPOVST ;R01/RMM Pop-Up for CHECK PATIENT VESTING in CPRS ;3/23/2012

;;3.0;ORDER ENTRY/RESULTS REPORTING;\*\*431\*\*;;Build 7

;268 NAME: DBIA268-A

;5408 NAME: CPT/HCPCS Procedure File 81

;IA 1625 NAME: PERSON CLASS API'S

;

;a patient is concidered vested if they have an inpatient admission or observation stay of less than 24 hours,

;or outpatient care that in general, equates to a primary care visit by a clinician authorized to administer a

;primary care visit. A primary care visit is identified by a list of specific Current Procedural Terminology

;(CPT) codes identified in this manual. These codes must be administered by at least one clinical provider

;authorized to complete the equivalent of a history and physical. The precise CPT codes and authorized providers

;are identified in the documentation of the Non-Vested patient class. A patient is required to meet the Vesting

;criteria once during the current year or the prior two fiscal years. Patients that do not meet the Vesting

;requirements are placed in the Non-Vested patient class.

EN(ORPODFN) ;CALCULATE VESTMENT

N ORPORET,ORPOBD,ORPOED,ORPOY,ORPOM,ORPOI,ORPOII,ORPOA

S ORPOY=$E(DT,1,3),ORPOM=$E(DT,4,5),ORPOY=$S(ORPOM>9:ORPOY-2,1:ORPOY-3),ORPOBD=ORPOY\_1001,ORPOED=DT,ORPORET="NON-VESTED"

;

;

S ORPOI="" F S ORPOI=$O(^OR(100.0074,"B",ORPOI)) Q:ORPOI="" D

.S:ORPOI["ICPT" ORPOA($$GET1^DIQ(81,$P(ORPOI,";"),.01))=""

.S:ORPOI["USC" ORPOA($$GET1^DIQ(8932.1,$P(ORPOI,";"),5))=""

D CVS(ORPODFN,.ORPORET)

Q ORPORET

CVS(ORPOPT,ORPOR) ;Calculates if a patient has the required local activity to be considered vested, within the current vesting period.

;The order of the search is local ward admission, fee basis inpatient activity, required cpt code in local outpatient activity,

;and required PERSON CLASS in fee basis outpatient activity.

;This function is looking for the first occurrence within the vesting period. Once an occurrence is found the hunt is over.

;The cpt codes used in the search are found in file 100.0074 and provider types defined as acceptable person classes

;are in file 100.0074

I $G(ORPOPT)="" S ORPOR="INVALID DFN" Q

Q:ORPOR="INVALID DFN"

I '$D(^DPT(ORPOPT)) S ORPOR="INVLAID DFN" Q

S:$$GET1^DIQ(2,ORPOPT,.152)]"" ORPOR="NOT ELIGIBLE" ;screen out patients not eligible

S:$$GET1^DIQ(2,ORPOPT,1901,"I")'="Y" ORPOR="NON-VETERAN" ;screen out non-veterans

Q:ORPOR'="NON-VESTED"

D ADM(ORPOPT,.ORPOR) Q:ORPOR="VESTED"

D FEE(ORPOPT,.ORPOR) Q:ORPOR="VESTED"

D FND(ORPOPT,.ORPOR)

Q

;D LIST^DIC(162.02,","\_15682\_","\_38728\_",","@;.01I;","Q",,,,"B")

;D LIST^DIC(162.02,","\_15682\_","\_38728\_",","@;.01I;","Q",,,,"B",,,"ORPO")

;D LIST^DIC(162.03,","\_2\_","\_15682\_","\_38728\_",","@;.01;","Q",,,,"B",,,"ORPO")

FND(ORPOPT,ORPOR) ;

N ORPOI,ORPOEP,ORPOFDT,ORPODT,ORPORN,ORPOPC,ORPOVN,ORPOII,ORPODOC

;^AUPNVCPT("AA",68,82435,7009871,3769349)=""

; PATIENT,CPT ,REVERSE DATE

;F ORPOI=0:0 S ORPOI=$O(ORPOA(ORPOI)) Q:ORPOI["V" D:$D(^AUPNVCPT("AA",ORPOPT,ORPOI)) ;visit xref in v cpt file

S ORPOI="" F S ORPOI=$O(ORPOA(ORPOI)) Q:ORPOI["V"!(ORPOI="") D:$D(^AUPNVCPT("AA",ORPOPT,ORPOI)) ;visit xref in v cpt file

.S ORPODT=9999999-(ORPOED+1) F S ORPODT=$O(^AUPNVCPT("AA",ORPOPT,ORPOI,ORPODT)) Q:'ORPODT!(ORPODT>(9999999-ORPOBD)) D

..S ORPORN=$O(^AUPNVCPT("AA",ORPOPT,ORPOI,ORPODT,0)),ORPOVN=$$GET1^DIQ(9000010.18,ORPORN,.03,"I") ;visit ien

..S ORPOEP=$$GET1^DIQ(9000010.18,ORPORN,1204,"I") S:ORPOEP]"" ORPODOC(ORPOEP)="" ;v cpt file encounter provider

..;D:('ORPOEP)&(ORPOVN) ;if no provider, but visit ien

..D:ORPOVN ;if visit ien

...D FIND^DIC(9000010.06,,"@;.01I;.04I","Q",ORPOVN,,"AD") ;v provider file

...F ORPOII=0:0 S ORPOII=$O(^TMP("DILIST",$J,2,ORPOII)) Q:'ORPOII D

....S ORPOEP=^TMP("DILIST",$J,"ID",ORPOII,.01) ;visit provider

....Q:'ORPOEP ;no encounter provider for the cpt code

....S ORPODOC(ORPOEP)="" ;encounter provider

..F ORPOEP=0:0 S ORPOEP=$O(ORPODOC(ORPOEP)) Q:'ORPOEP D

...S ORPOFDT=9999999-ORPODT,ORPOPC=$$GET^XUA4A72(ORPOEP,ORPOFDT)

...Q:ORPOPC=-1 ;not a valid user or person class never assigned

...Q:ORPOPC=-2 ;no active person class on that date

...Q:$P(ORPOPC,U,7)="" ;QUIT IF NO DANCSD

...S:$D(ORPOA($P(ORPOPC,U,7))) ORPOR="VESTED"

Q

ADM(ORPOPT,ORPOR) ;IF ADMITTED IN LAST TWO YEARS VESTED

;R01/RMM \*\*\*MODIFICATION\*\*\* 8/14/2015

;MODIFIED TO FIND ALL INPATIENTS DURNING VESTING PERIOD

;THE OLD CODE ONLY FOUND PATIENTS ADDMITED DURING THE VESTING PERIOD

;N ORPOAD

;S ORPOAD=ORPOBD F S ORPOAD=$O(^DGPM("APTT1",ORPOPT,ORPOAD)) Q:'ORPOAD!(ORPOAD>(ORPOED+.9999)) D

;.S ORPOR="VESTED"

;

D FIND^DIC(45,,"@;.01I;2I;11;13I","Q",ORPOPT,,"B") ;PTF FILE

N ORPOI,ORPOF S ORPOF=0

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI!(ORPOF=1) D

.Q:^TMP("DILIST",$J,"ID",ORPOI,.01)'=ORPOPT

.S:^TMP("DILIST",$J,"ID",ORPOI,2)>ORPOBD ORPOR="VESTED",ORPOF=1

.D:^TMP("DILIST",$J,"ID",ORPOI,11)="CENSUS"

..D:^TMP("DILIST",$J,"ID",ORPOI,13)]""

...S:$$GET1^DIQ(45.86,^TMP("DILIST",$J,"ID",ORPOI,13),.01,"I")>ORPOBD ORPOR="VESTED",ORPOF=1

;\*\*\* END MODIFICATION \*\*\*

Q

;

FEE(ORPOPT,ORPOR) ; FEE BASIS PATIENT

N ORPOFP,ORPOTD,ORPOI,ORPOII,ORPOIII,ORPOC,ORPOLST,ORPOFDT,ORPOF,ORPOD3,ORPOVEN S ORPOF=0

S ORPOFP=0 F S ORPOFP=$O(^FBAAA("AQLVS",ORPOPT,ORPOFP)) Q:'ORPOFP D

.Q:ORPOFP=2 ;screen out the outpatient fee basis program

.S ORPOTD=ORPOBD-1 F S ORPOTD=$O(^FBAAA("AQLVS",ORPOPT,ORPOFP,ORPOTD)) Q:'ORPOTD D

..S ORPOR="VESTED"

Q:ORPOR="VESTED"

D LIST^DIC(162.01,","\_ORPOPT\_",","@;.01I;","Q",,,,"B") ;FEE BASIS PAYMENT PAYMENT

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI!(ORPOF=1) D

.S ORPOVEN=^TMP("DILIST",$J,"ID",ORPOI,.01)

.D LIST^DIC(162.02,","\_ORPOVEN\_","\_ORPOPT\_",","@;.01I;","Q",,,,"B",,,"ORPOLST")

.F ORPOII=0:0 S ORPOII=$O(ORPOLST("DILIST",2,ORPOII)) Q:'ORPOII!(ORPOF=1) D

..S ORPOFDT=ORPOLST("DILIST","ID",ORPOII,.01)

..Q:ORPOBD>ORPOFDT ;SCREEN OUT IF BEGINNING DATE IS AFTER DT

..Q:ORPOED<ORPOFDT ;SCREEN OUT IF END DATE IS BEFORE DT

..S ORPOD3=ORPOLST("DILIST",2,ORPOII)

..D LIST^DIC(162.03,","\_ORPOD3\_","\_ORPOVEN\_","\_ORPOPT\_",","@;.01;","Q",,,,"B",,,"ORPOC")

..F ORPOIII=0:0 S ORPOIII=$O(ORPOC("DILIST",2,ORPOIII)) Q:'ORPOIII D

...S:$D(ORPOA(ORPOC("DILIST","ID",ORPOIII,.01))) ORPOR="VESTED",ORPOF=1

Q

ADDT(X1,X2) ;ADD/SUBTRACT FROM DATE

N X D C^%DTC

Q X

PRTV ;ENTRY POINT FOR PRINTING VESTING CODES

K ZTSAVE D EN^XUTMDEVQ("START^ORPOVST","ORPOOR PRINT VESTING CODES")

Q

START ;ENTRY POINT

K ^TMP("DILIST",$J),^TMP("ORPOORUTL",$J)

D LIST^DIC(100.0074,,"@;.01I","Q",,,,"B")

D GPTP,GCPT

K ^TMP("DILIST",$J),^TMP("ORPOORUTL",$J)

Q

GPTP ;GET PROVIDER TYPE

N ORPOI,ORPOIEN,ORPO

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI D ;PRINT DANCSD S

.Q:^TMP("DILIST",$J,"ID",ORPOI,.01)["ICPT"

.S ORPOIEN=$P(^TMP("DILIST",$J,"ID",ORPOI,.01),";")

.K ORPO

.D FIND^DIC(8932.1,,"@;5;6;","Q","`"\_ORPOIEN,,,,,"ORPO")

.S ^TMP("ORPOORUTL",$J,ORPO("DILIST","ID",1,5))=ORPO("DILIST","ID",1,6)

D PTPV

Q

GCPT ;GET PROVIDER TYPE

N ORPOI,ORPOIEN,ORPOCPT K ^TMP("ORPOORUTL",$J)

F ORPOI=0:0 S ORPOI=$O(^TMP("DILIST",$J,2,ORPOI)) Q:'ORPOI D ;PRINT DANCSD S

.Q:^TMP("DILIST",$J,"ID",ORPOI,.01)["USC"

.S ORPOIEN=$P(^TMP("DILIST",$J,"ID",ORPOI,.01),";")

.S ORPOCPT=$$GET1^DIQ(81,ORPOIEN,.01)

.S ^TMP("ORPOORUTL",$J,ORPOCPT)=""

D PTCPT

Q

PTCPT ;PRINT CPT

N ORPOI,ORPOCOL,ORPOF,ORPORET S ORPOCOL=0,ORPORET="!",ORPOF=0

W !!!!,"CPT codes for CPRS vesting",?40,"Print Date: ",$$CNVDT(DT),!

S ORPOI="" F S ORPOI=$O(^TMP("ORPOORUTL",$J,ORPOI)) Q:ORPOI="" D

.I ORPORET]"" W @ORPORET,?ORPOCOL,ORPOI

.E W ?ORPOCOL,ORPOI

.S:ORPOCOL=60 ORPOCOL=0,ORPORET="!",ORPOF=1

.S:ORPOCOL=50 ORPOCOL=60,ORPORET=""

.S:ORPOCOL=40 ORPOCOL=50,ORPORET=""

.S:ORPOCOL=30 ORPOCOL=40,ORPORET=""

.S:ORPOCOL=20 ORPOCOL=30,ORPORET=""

.S:ORPOCOL=10 ORPOCOL=20,ORPORET=""

.I ORPOCOL=0,ORPOF=0 S ORPOCOL=10,ORPORET=""

.S ORPOF=0

Q

CNVDT(Y) ;FORMAT INTERNAL TO EXTERNAL DATE

D DD^%DT

Q Y

PTPV ;PRINT PROVIDER TYPE

N ORPOI,ORPOCOL,ORPOF,ORPORET S ORPOCOL=0,ORPORET="!",ORPOF=0

W !!!,"Provider Types for CPRS vesting",?45,"Print Date: ",$$CNVDT(DT),!

S ORPOI="" F S ORPOI=$O(^TMP("ORPOORUTL",$J,ORPOI)) Q:ORPOI="" D

.I ORPORET]"" W @ORPORET,?ORPOCOL,ORPOI," - ",^TMP("ORPOORUTL",$J,ORPOI)

.E W ?ORPOCOL,ORPOI," - ",^TMP("ORPOORUTL",$J,ORPOI)

.S:ORPOCOL=52 ORPOCOL=0,ORPORET="!",ORPOF=1

.S:ORPOCOL=26 ORPOCOL=52,ORPORET=""

.I ORPOCOL=0,ORPOF=0 S ORPOCOL=26,ORPORET=""

.S ORPOF=0

Q

;

;

;

TEST(DT) ;\*\*\*\*\* code used for testing new dates \*\*\*\*\*

;The transition to the 2-year rolling population will occur incrementally over the course of three consecutive VERA Models.

;Each of the three impending models are listed below

;VERA 2015: Fund 2.66 years of Basic Care population (i.e. reduce third year population by 33%)

;VERA 2016: Fund 2.33 years of Basic Care population (i.e. reduce third year population by 66 %)

;VERA 2017: Fund rolling 2-year Basic Care patient population.

N ASV,ASV1

F ASV=1:1:9 D ;year

.F ASV1=1:1:12 D ;month

..S DT=$E(DT,1,2)\_ASV\_$S($L(ASV1)=2:ASV1,1:"0"\_ASV1)\_15 ; W !,DT

..S ORPOY=$E(DT,1,3),ORPOM=$E(DT,4,5),ORPOY=$S(ORPOM>9:ORPOY-2,1:ORPOY-3),ORPOBD=ORPOY\_1001,ORPOED=DT,ORPORET="NON-VESTED"

..W !!,DT,?16,ORPOBD,?30,$$CNVDT(ORPOBD)

..S (ORPOY,ORPOM,ORPOBD)=""

..S ORPOY=$E(DT,1,3),ORPOM=$E(DT,4,5),ORPOED=DT,ORPORET="NON-VESTED" S:ORPOM>9 ORPOY=ORPOY+1 ;,ORPOBD=ORPOY\_1001,

..D:ORPOY<315

...S ORPOY=ORPOY-3,ORPOBD=ORPOY\_1001

..D:ORPOY=315

...S ORPOY=ORPOY-2,ORPOBD=ORPOY\_"0201"

..D:ORPOY=316

...S ORPOY=ORPOY-2,ORPOBD=ORPOY\_"0601"

..D:ORPOY>=317

...S ORPOY=ORPOY-2,ORPOBD=ORPOY\_1001

..W !,$$CNVDT(DT),?16,ORPOBD,?30,$$CNVDT(ORPOBD)

Q