Clinical Reminders Index

Technical Manual

February 2016

Provider Systems
Office of Information and Technology
Department of Veterans Affairs

Revision History

Date	Page #	Description	Developer/Technical Writer
February 2016	17, 22, 25	Change to V Immunization index, per PX*1.0*210	Levi Teitelbaum/ Kathy Steele/ Shelita Davis
May 2014		Many updates per Developer review	Patrick Redington/ JoAnn Green
Feb 2014	5	Overview of updates per the Reminders ICD-10 Update project	Patrick Redington/ JoAnn Green
Feb 2014	19	Change to Problem List index, per GMPL*2.0*44	Patrick Redington/ JoAnn Green
Feb 2014	19	Change to Registration index, per DG*5.3*862	Patrick Redington/ JoAnn Green
May 2012		Corrected codes in Registration Index text	Patrick Redington/ JoAnn Green
Oct 2008	17	Updated MH Index	Patrick Redington/ JoAnn Green
Dec 2009	22	Update MH Index in Summary Table	Patrick Redington/ JoAnn Green
Dec 2004		Original release of Reminders Index (PXRM*1*12)	Patrick Redington/ JoAnn Green

Contents

Clinical Reminders Index Global

Introduction

The Clinical Reminders Index global has been designed to provide an index of clinical data, which supports rapid access to clinical data. It is used by Clinical Reminders v2.0, which evaluates reminders in 1/3 to 1/2 the time that v1.5 required. A large part of the speedup can be attributed to the Index, because it provides such an efficient way to find patient data. The Index is a resource that can be used by other packages or site-developed applications whenever they need to find patient data. Use of the Index is supported by subscription to ICR #4290. This document describes the structure of the Index and how to use it, as well as how to populate it and manage it.

The basic structure of the index is:
^PXRMINDX(FILE NUMBER,”IP”,ITEM,DFN,DATE,DAS)
^PXRMINDX(FILE NUMBER,”PI”,DFN,ITEM,DATE,DAS)

where “IP” stands for item and patient and “PI” for patient and item.

The “IP” index lets you find all patients with a particular item, while the “PI” index lets you find all items for a patient. DAS stands for DA string, similar to FileMan’s DA array. It is a semicolon-separated string that specifies the exact location in the source global where the data is stored. This can be as simple as the internal entry number (IEN) or can have a number of pieces (e,g,, a lab test result has four pieces).

Indexes with items that are coded values, such as ICD codes, vary from the basic structure, in order to support look-ups based on data such as primary diagnosis or principal procedure. Details are found in the sections describing each index.

Global Placement

This global serves as an index for the clinical data in a number of packages, so it is independent of a particular package. When new data is entered into the globals being indexed, the Index will grow, so it needs to be placed where it has room to grow. There is a utility for estimating the initial size of the Index. To run this utility at the programmer prompt, type D ESTTASK^PXRMISE. This will start a TaskMan job that estimates the initial size of the index for each global as well as the total size. The information will be delivered in a MailMan message sent to members of the mail group defined in file #800. The estimated sizes will be given as a number of blocks which are 2K for Caché sites.

We recommend placing ^PXRMINDX in its own volume set. The initial size of the dataset should be based on the estimated size, plus leaving room for growth.

Journaling
Journaling of PXRMINDX is NOT required, because the index can be rebuilt, if necessary.

Changes to Reminder Indexes made by the Clinical Reminders ICD-10 Update project

To support the introduction of ICD-10 codes into VistA, Clinical Reminders has taken a very general approach, wherein Clinical Reminders taxonomies are being restructured to be Lexicon-based instead of pointer-based. This allows the use of any coding system supported by the Lexicon package. In addition to adding ICD-10 codes, SNOMED CT codes are being added. With the release of CPRS 29, SNOMED CT codes can be collected by Problem List and Clinical Reminders will be able to search for them.

The sources of coded data that are indexed are Problem List, PTF, V CPT, and V POV. For these files, with some exceptions, the first two subscripts of the Index will become
^PXRMINDX(FILE NUMBER,CODING SYSTEM)
Where coding system is the standard three-character Source Abbreviation defined in file #757.03. The exceptions apply to V CPT and V POV and are described in the PCE section.

Patches in the Clinical Reminders project build:

DG*5.3*862
This build updates the Clinical Reminders Index cross-references in the PTF file (#45) to accommodate ICD-10 diagnosis and procedure codes. It restructures the PTF portion of the Clinical Reminders Index to a generic format that can support all ICD coding systems. This format is:

```
^PXRMINDX(45,CODING SYSTEM,"INP",CODE,NODE,DFN,DATE,DAS)
^PXRMINDX(45,CODING SYSTEM,"PNI",DFN,NODE,CODE,DATE,DAS)
```

where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer.

The post-install routine will start a background job to rebuild the file #45 index in the new format.

GMPL*2.0*44

This build updates the Clinical Reminders Index cross-references in the Problem file (#9000011) to accommodate ICD-10 CM diagnosis codes. It restructures the Problem List portion of the Clinical Reminders Index to a generic format that can support ICD and SNOMED CT coding systems. This format is:

```
^PXRMINDX(9000011,CODING SYSTEM,”ISPP”,CODE,STATUS,PRIORITY,DFN,DLM,DAS)
^PXRMINDX(9000011,CODING SYSTEM,”PSPI”,DFN,STATUS,PRIORITY,CODE,DLM,DAS)
```

where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. The post-install routine will start a background job to rebuild the file #9000011 index in the new format.

Clinical Reminders Index Management

The option, PXRM INDEX MANAGEMENT, is a menu containing PXRM INDEX BUILD and PXRM INDEX COUNT.

The index building utility serves two purposes:

1. It initially populates the indexes by indexing the existing data. It works its way through the entire global, putting entries in the index for each piece of unique patient data it finds.
2. If the index ever gets corrupted or destroyed, the utility can be used to rebuild the index. Therefore, there is no need to journal the index, since it can be recreated from the original data at any time.

When the index utility finishes indexing a particular global, it sets the following three nodes:

```
^PXRMINDX(FILE NUMBER,”GLOBAL NAME”)=$$GET1^DID(FILE NUMBER,””,””,”GLOBAL NAME”)
^PXRMINDX(FILE NUMBER,”BUILT BY”)=DUZ
^PXRMINDX(FILE NUMBER,”DATE BUILT”)=$$NOW^XLFDT
```

In addition to providing information about who built the index and when it got populated, these nodes can be used to determine when the index is complete and ready for use.

When data is added, edited, or deleted, the indexes are kept current using new-style MUMPS cross-references. These cross-references call APIs that set and kill the index entries.

PXRM INDEX BUILD Option

The index build utility can be accessed through the option PXRM INDEX MANAGEMENT or directly via PXRM INDEX BUILD.

```
Select OPTION NAME: PXRM INDEX BUILD       PXRM INDEX BUILD     run
routine
PXRM INDEX BUILD
Which indexes do you want to (re)build?
  1 - LABORATORY TEST (CH, Anatomic Path, Micro)
  2 - MENTAL HEALTH
  3 - ORDER
  4 - PTF
  5 - PHARMACY PATIENT
  6 - PRESCRIPTION
  7 - PROBLEM LIST
  8 - RADIOLOGY
  9 - V CPT
 10 - V EXAM
 11 - V HEALTH FACTORS
 12 - V IMMUNIZATION
 13 - V PATIENT ED
 14 - V POV
 15 - V SKIN TEST
 16 - VITAL MEASUREMENT
Enter your list:  (1-16): 1-16
```

It can be used to build or rebuild the indexes for each of the globals, in any combination you want. You can run the build interactively or as a tasked job. Rebuilding an index is not something you would normally want to do. If problems are found with some entries in an index it is better to deal with them on an individual basis. Rebuilding an entire index is a last resort when there is no other way to repair or restore it.

After selecting the globals to be indexed, you will be given the choice of submitting a tasked job or running it interactively. In either case, when the index build completes, members of the mail group defined in file #800 will be sent a MailMan message that looks something like:

```
Subj: Index for global AUPNVPED successfully built  [#12184]
10/03/02@10:25  5 lines
From: POSTMASTER (Sender: DOE,JOHN)  In 'IN' basket.   Page 1  *New*
-----------------------------------------------------------------------
Build of Clinical Reminders index for global AUPNVPED completed.
Build finished at 10/03/2002@10:25:27
288 entries were created.
Elapsed time: 1 secs
0 errors were encountered.
```

NOTE: If the person who builds the indexes is not a member of the mail group, the messages will not be delivered to the members of the mail group, unless it is a public mail group.

For large globals, the build time can be many hours, so you may want to schedule these builds for overnight or weekends when the system is not under heavy use. The indexes for smaller globals, which require few block splits, can be built quickly (a few minutes), so you have wider latitude in when to build those.

Error Messages

If any entries couldn’t be indexed, the completion message will look something like:

```
Subj: Index for global PS(55 successfully built  [#12187]
10/03/02@10:30
6 lines
From: POSTMASTER (Sender: DOE,JOHN)  In 'IN' basket.   Page 1  *New*
-----------------------------------------------------------------------
Build of Clinical Reminders index for global PS(55 completed.
Build finished at 10/03/2002@10:30:07
6416 entries were created.
Elapsed time: 13 secs
```

```
136 errors were encountered.
```

Another MailMan message contains the error information for up to the last N errors. The error information starts with the most recent entry in the global that has an error and progressively works back toward older entries. The number of errors displayed, N, is a site-configurable parameter. The parameter is stored in the CLINICAL REMINDERS PARAMETERS file, #800, in the field MAXIMUM NUMBER OF INDEX ERRORS, field #15 of file #800, Clinical Reminder Parameters. The default value is 200. If you wish to change this, you can use the ENTER OR EDIT FILE ENTRIES FileMan option. If you find a substantial number of errors, it is likely there is a systematic problem, so after determining the cause and solution for the first few errors, you can probably apply the same corrective action to the bulk of them.

The message has the format: global, entry identification, and error message. The error message describes the problem – for example, missing or invalid data. Here is a sample of errors you might see for file #55:

```
Subj: CLINICAL REMINDER INDEX BUILD ERROR(S)  [#14895] 11/19/02@11:50  136
lines

From: POSTMASTER (Sender: DOE,JOHN)  In 'IN' basket.   Page 1
-------------------------------------------------------------------------------
GLOBAL: PS(55 ENTRY: DFN=1 D1=33 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=34 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=35 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=1 D1=36 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=2 D1=1 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=1719 Unit Dose missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=1 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=2 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=3 D1=3 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=4 D1=2 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=4 D1=8 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=5 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=7 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=6 D1=590 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=8 D1=1 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=9 D1=2 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=9 D1=7 D2=1 IV additive missing drug
GLOBAL: PS(55 ENTRY: DFN=10 D1=31 IV missing stop date
GLOBAL: PS(55 ENTRY: DFN=10 D1=32 IV missing stop date
```

The entry information identifies the exact entry in the global that could not be indexed. If we examine the last line, it tells us that for ^PS(55,10,”IV”,32,0), there is no stop date, so it can’t be indexed. If you are not familiar with the structure of the global being indexed, it will be helpful to have a data dictionary listing on hand to help you interpret the entry identification information.

Entries that are not indexable will never be found by any application that uses the index to find data. Each site should make a decision concerning what they want to do about non-indexable entries.

Inpatient Pharmacy

Sites have reported that the bulk of the errors are the same type: missing start date or missing patient.

.

```
GLOBAL: ^PS(55, ENTRY: DFN=994 D1=1545755 Unit Dose missing start date
Global ^PS(55,994,,1545755
        PS(55,994,,1545755
^PS(55,994,5,1545755,0) = ^^^^^^^^E
^PS(55,994,5,1545755,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=388 D1=1804771 Unit Dose missing start date
Global ^PS(55,388,,1804771
        PS(55,388,,1804771
^PS(55,388,5,1804771,0) = ^^^^^^^^E
^PS(55,388,5,1804771,9,0) = ^55.09D^1^1

GLOBAL: ^PS(55, ENTRY: DFN=1096 D1=1819001 Unit Dose missing start date
Global ^PS(55,1096,,1819001
        PS(55,1096,,1819001
^PS(55,1096,5,1819001,0) = ^^^^^^^^E
^PS(55,1096,5,1819001,9,0) = ^55.09D^1^1
Global ^

GLOBAL: ^PS(55, ENTRY: DFN=1245 D1=16 IV missing start date
Global ^PS(55,1245,,16
        PS(55,1245,,16
^PS(55,1245,"IV",16,0) = 16^^^P^^^^INFUSE VIA
MINIPUMP^NOW^^1000^^^^0^^S X=PSSTA
TUS
^PS(55,1245,"IV",16,1) = ONE TIME
^PS(55,1245,"IV",16,2) = 2861123.1008^1^IBG
^PS(55,1245,"IV",16,3) = PROTECT FROM LIGHT/DO NOT REFRIGERATE
^PS(55,1245,"IV",16,6) = 67^20 MG
^PS(55,1245,"IV",16,"A",0) = ^55.04A^1^1
^PS(55,1245,"IV",16,"A",1,0) = 1^D^IBG^COMPUTER DOWN^2861123.1045
^PS(55,1245,"IV",16,"A",2,1,0) = ^55.15^1^1
^PS(55,1245,"IV",16,"A",2,1,1,0) = STATUS^DISCONTINUED^
^PS(55,1245,"IV",16,"AD",0) = ^55.02PA^1^1
^PS(55,1245,"IV",16,"AD",1,0) = 108^20 MG
^PS(55,1245,"IV",16,"SOL",0) = ^55.11IPA^1^1
^PS(55,1245,"IV",16,"SOL",1,0) = 1^10 ML
```

Outpatient Pharmacy

Sites have reported that the bulk of the errors are the same type: missing drug or missing patient.

```
GLOBAL: ^PSRX( ENTRY: 200167 missing drug^PSRX(200167,0)="
^154^^^^^^^^^^^2881118^^^^^^1"
             2)="2881118^2881127^^^^^^^1^ "
             3)=2881127
         "POE")=1
         "SIG")="^0"
         "STA")=11
        "TYPE")=0
```

```
GLOBAL: ^PSRX( ENTRY: 5287379 missing DFN
Global ^PSRX(5287379
        PSRX(5287379
^PSRX(5287379,0) = 5285540
^PSRX(5287379,2) = ^^^^^^^^^^
^PSRX(5287379,3) =
^PSRX(5287379,"POE") = 1

GLOBAL: ^PSRX( ENTRY: 5288355 missing DFN
Global ^PSRX(5288355
        PSRX(5288355
^PSRX(5288355,0) = 5285807
^PSRX(5288355,2) = ^^^^^^^^^^
^PSRX(5288355,3) =
^PSRX(5288355,"POE") = 1
```

Disable/Enable Reminder Evaluation

The option PXRM INDEX BUILD provides the ability to rebuild selected portions of the Clinical Reminders Index. While an index is rebuilding, any reminder that uses the data from that index cannot be correctly evaluated – it will have the status of CNBD (cannot be determined). In the past, a MailMan message was sent to the Clinical Reminders mail group every time a reminder could not be evaluated because an index was rebuilding. Now, when an index is going to be rebuilt, reminder evaluation will be automatically disabled, meaning that any attempt to evaluate a reminder will result in an immediate return of a CNBD status. The Clinical Maintenance display will include text letting the user know that reminder evaluation is disabled and the reason(s). When the index has finished rebuilding, evaluation will be automatically enabled.

The option PXRM DISABLE/ENABLE EVALUATION provides a manual disable/enable function. If for some reason, reminder evaluation needs to be disabled, it can be done through this option. This option should be given to a very limited number of people and can only be used by holders of the PXRM MANAGER key. When the issue that required disabling evaluation has been handled, reminder evaluation can be enabled again using this same option. Note that this option can be used to enable evaluation even if it was not disabled using this option. For example, if reminder evaluation was automatically disabled for an index rebuild, this option could be used to enable evaluation even though the index is still rebuilding. If that is done, the MailMan messages will start being sent again.

When reminder evaluation is disabled, the following options and protocols will be put out of order.

Options:
 PXRM DEF INTEGRITY CHECK ALL
 PXRM DEF INTEGRITY CHECK ONE
 PXRM ORDER CHECK TESTER
 PXRM REMINDERS DUE
 PXRM REMINDERS DUE (USER)

Protocols:
 PXRM PATIENT LIST CREATE
 PXRM EXTRACT MANUAL TRANSMISSION

When reminder evaluation is again enabled, these options and protocols will be put back in order.

Anytime reminder evaluation is disabled, a message with the subject “REMINDER EVALUATION DISABLED” will be sent to the Clinical Reminders mail group. The message will give the date and time that evaluation was disabled, list the reasons for disabling evaluation, and a search will be made for any Clinical Reminders TaskMan jobs that could be affected. There will be a list of those that are found; it will include the job description, the status (pending or running), and the task number. The results of any jobs that are already running will be unreliable and should be discarded. If possible, these jobs should be stopped, so that they don’t waste system resources. None of the pending jobs should be allowed to start until evaluation is enabled again.

When evaluation is enabled, a message with the subject “REMINDER EVALUATION ENABLED” will be sent to the Clinical Reminders mail group. It will contain the date and time evaluation was disabled and when it was enabled. This gives you the exact period of when evaluation was disabled.

Here are examples of disable and enable messages:

```
MailMan message for CRMANAGER,TWO
Printed at CPRS30.FO-SLC.MED.VA.GOV  04/16/14@10:32
Subj: REMINDER EVALUATION DISABLED  [#122941] 04/16/14@10:30  58 lines
From: POSTMASTER (Sender: CRMANAGER, ONE)  In 'IN' basket.   Page 1
-------------------------------------------------------------------------------
Reminder evaluation was disabled on Apr 16, 2014@10:30:42.
Because of this, the following TaskMan jobs can produce erroneous results.
Pending jobs should not be allowed to start until evaluation is enabled.
The results of running jobs should be discarded and if possible, running jobs
should be stopped.

Reason: index rebuild for file #45.

 Reminders Due Report Jobs

Task number - 316820
Status - Active: Running
Time - Feb 08, 2012@12:40:28
User - CRCOORDINATOR, TWO

 Reminder Patient List Jobs

Task number - 1980022
Status - Active: Running
Time - Apr 16, 2014@08:00
User - TASKMAN,PROXY USER
```

```
Task number - 1980207
Status - Active: Pending
Time - Apr 17, 2014@08:00
User - TASKMAN,PROXY USER

 Reminder Extract Jobs

Task number - 342256
Status - Active: Pending
Time - Mar 06, 2012@20:04:25
User - CRCOORDINATOR, SIX

Task number - 1867565
Status - Active: Pending
Time - May 17, 2013@16:44:13
User - CRCOORDINATOR, SIX

Task number - 1902474
Status - Active: Pending
Time - Jul 17, 2013@17:16:30
User - CRCOORDINATOR,TEN

Task number - 1945932
Status - Active: Pending
Time - Oct 22, 2013@12:37:23
User - CRCOORDINATOR, THIRTY

Task number - 1946204
Status - Active: Pending
Time - Oct 23, 2013@12:37:35
User - CRCOORDINATOR, TWO

Task number - 1966964
Status - Active: Pending
Time - Feb 05, 2014@07:54:39
User - CRCOORDINATOR,THREE

Enter message action (in IN basket): Ignore//


Subj: REMINDER EVALUATION ENABLED  [#122942] 04/16/14@10:30  2 lines
From: POSTMASTER (Sender: CRMANAGER, ONE)  In 'IN' basket.   Page 1
-------------------------------------------------------------------------------
Reminder evaluation was enabled on Apr 16, 2014@10:30:49.
It was disabled on Apr 16, 2014@10:30:42.

Enter message action (in IN basket): Ignore//
```

PXRM INDEX COUNT Option

The index count utility can be accessed through the option PXRM INDEX MANAGEMENT or directly via PXRM INDEX COUNT. This utility provides a count by year of the entries in the index. This will let sites see how their data is distributed on a yearly basis.

The selection for the count utility is identical to the build utility and the results are sent in a MailMan message just like the results of the build utility.

Index Details

Inpatient Pharmacy

The index is on file 55, Pharmacy Patient. The structure is:

```
 ^PXRMINDX(55,”IP”,DRUG,DFN,START,STOP,DAS)
 ^PXRMINDX(55,”PI”,DFN,DRUG,START,STOP,DAS)
```

where DRUG is a pointer to the Drug file (#50), START is the start date, and STOP is the stop date. The API to retrieve the associated data is OEL^PSJPXRM1(DAS,.DATA).

It is documented in ICR #3836.

Non-VA Meds

The index is on the non-VA med multiple of file #55, Pharmacy Patient. The structure is:

```
^PXRMINDX(“55NVA”,”IP”,POI,DFN,START,STOP,DAS)
 ^PXRMINDX(“55NVA”,”PI”,DFN,POI,START,STOP,DAS)
```

where POI is a pointer to the Pharmacy Orderable Item file. START is the start date, if it exists; otherwise it is the documented date. STOP is the discontinued date if it exists; otherwise is “U”_D0. If STOP has the form “U”_D0 it should be interpreted to mean that no discontinued date exists so the patient is currently taking the non-VA med.

The API to retrieve the associated data is NVA^PSOPXRM1(DAS,.DATA). It is documented in ICR 3793.

Outpatient Pharmacy

The index is on file #52, Prescription file. The structure is:

```
 ^PXRMINDX(52,”IP”,DRUG,DFN,START,STOP,DAS)
 ^PXRMINDX(52,”PI”,DFN,DRUG,START,STOP,DAS)
```

where DRUG is a pointer to Drug file, START is the starting date (RELEASE DATE) and STOP is the stop date (RELEASE DATE + DAYS SUPPLY). The API to retrieve the associated data is PSRX^PSOPXRM1(DAS,.DATA).

It is documented in ICR #3793.

Order Entry

The index is on file 100, Order file. The structure is:

```
 ^PXRMINDX(100,”IP”,OI,DFN,START,STOP,DAS)
 ^PXRMINDX(100,”PI”,DFN,OI,START,STOP,DAS)
```

where OI is a pointer to the Orderable Items file, START is the START DATE, and STOP is the STOP DATE. The API to retrieve the associated data is EN^ORX8(DA). Note that DA is the first piece of DAS and the data is returned in the array ORUPCHK.

The API is documented in ICR #871.

Lab
The index is on file 63, Lab Data. The structure is:

Chemistry, Hematology, other Lab Tests

```
 ^PXRMINDX(63,”IP”,ITEM,DFN,DATE,DAS)
 ^PXRMINDX(63,”PI”,DFN,ITEM,DATE,DAS)
```

Microbiology and Anatomic Path data have an additional index
^PXRMINDX(63,”PDI”,DFN,DATE,ITEM,DAS)
where DATE is the Date/Time of collection. The structure of ITEM depends on the type of lab data.

For Chemistry, Hematology, and other lab tests, ITEM is numeric and a pointer to the Laboratory Test file.

For Microbiology, ITEM is of the format
"M;[S T O A M];#".
where the middle section can be one of:
S is specimen (# is a pointer to the Topography [SNOMED] file)
T is test (# is a pointer to the Laboratory Test file)
O is organism (# is a pointer to the Etiology Field [SNOMED] file)
A is antibiotic (# is a pointer to the Antimicrobial Susceptibility file)
M is a TB drug (# is the field number of the TB drug - ^DD(63.39,).

For Anatomic Pathology, ITEM is of the format
"A;[S T O D M E F P I];#".
where the middle section can be one of:
S is specimen (# is 1.free text value)
T is test (# is a pointer to the Laboratory Test file)
O is organ/tissue (# is a pointer to the Topography [SNOMED] file)
D is disease (# is a pointer to Disease Field [SNOMED] file)
M is morphology (# is a pointer to the Morphology Field [SNOMED] file)
E is etiology (# is a pointer to the Etiology Field [SNOMED] file)
F is function (# is a pointer to the Function [SNOMED] file
P is procedure (# is a pointer to the Procedure [SNOMED] file)
I is ICD (# is a pointer to the ICD DIAGNOSIS file)

Microbiology and Anatomic Pathology data are stored in a complex hierarchy. The ITEM is therefore a compound expression. This allows single elements of data to be easily found. The DAS also depends on the type of lab data. A chemistry test result has four semicolon pieces. Microbiology and Anatomic Pathology can be more complex and have a much more nested structure.

The API to retrieve the associated data is LRPXRM^LRPXAPI(.DATA,DAS,ITEM). This information should be reviewed in the context of other data associated with the specimen. The API is documented in ICR #4245. The Lab package has other APIs that use these indexes.

Mental Health

The index is on file 601.84, MH Administrations. The structure is:

```
 ^PXRMINDX(601.84,”IP”,INS,DFN,DATE,DAS)
 ^PXRMINDX(601.84,”PI”,DFN,INS,DATE,DAS)
```

where INS is a pointer to the MH Instrument file #601. The API to retrieve the associated data is D ENDAS71^YTQPXRM6(.DATA,DAS)

The API is documented in ICR #5043.

.

PCE

There are indexes on all of the V files, with the exception of V Provider and V Treatment. There are two types of indexes for the V files – one for coded values and one for the non-coded values.

The non-coded values are stored in the following V files:

V FILE	FILE NUMBER
V EXAM	9000010.13
V HEALTH FACTORS	9000010.23
V IMMUNIZATION	9000010.11
V PATIENT ED	9000010.16
V SKIN TEST	9000010.12

The structure of the index for the non-coded values is:

```
 ^PXRMINDX(FILE NUMBER,”IP”,ITEM,DFN,DATE,DAS)
 ^PXRMINDX(FILE NUMBER,”PI”,DFN,ITEM,DATE,DAS)
```

where item is the .01 of the V file (for example, a pointer to the Education Topic file or Health Factor file), and DATE is the Visit Date. For V Immunization, if the Event Date and Time field (#1201) is populated, that will be used as the DATE instead of the Visit Date.

The V Immunization file has an additional index:

```
 ^PXRMINDX(9000010.11,"CVX","IP",CVX_CODE,DFN,DATE,DAS)
 ^PXRMINDX(9000010.11,"CVX","PI",DFN,CVX_CODE,DATE,DAS)
```

where DATE is the Event Date and Time. If Event Date and Time is not populated, the Visit Date/Time will be used instead. CVX is the Center for Disease Control’s vaccine administered code.

Note: The “ACR” cross-reference on the Immunization file updates the ^PXRMINDX(9000010.11,"CVX") index when an immunization’s CVX code is changed.

Coded values are stored in the following V files:

V FILE	FILE NUMBER	CODE TYPE
V CPT	9000010.18	CPT-4
V POV	9000010.07	ICD diagnosis

Because of the large number of entries, the existing structure of the Index for ICD-9 diagnosis and CPT-4 codes will be left as is:

```
 ^PXRMINDX(FILE NUMBER,”IPP”,CODEP,TYPE,DFN,DATE,DAS)
 ^PXRMINDX(FILE NUMBER,”PPI”,DFN,TYPE,CODEP,DATE,DAS)
```

where CODEP is a pointer to the coded value, TYPE is primary procedure for V CPT and primary/secondary for V POV. DATE is the Visit Date.

Starting with the ICD-10 update the structure of the Index will become:

```
^PXRMINDX(FILE NUMBER,CODING SYSTEM,”IPP”CODE,TYPE,DFN,DATE,DAS)
^PXRMINDX(FILE NUMBER,CODING SYSTEM,”PPI”,DFN,TYPE,CODE,DATE,DAS)
```

where CODING SYSTEM is the three-character Source Abbreviation from file #757.03.

For example, the structure of the V POV Index for ICD-10 diagnosis codes is:

```
 ^PXRMINDX(9000010.07,”10D”,”IPP”,CODE,TYPE,DFN,DATE,DAS)
 ^PXRMINDX(9000010.07,”10D”,”PPI”,DFN,TYPE,CODE,DATE,DAS)
```

where CODE is the ICD-10 the code.

The APIs for retrieving the associated date are in routine PXPXRM. There is an entry point for each V file; these are listed in the following table. The APIs are documented in ICR #4250.

V FILE	PXPXRM ENTRY POINT
V CPT	VCPT(DAS,.DATA)
V EXAM	VXAM(DAS,.DATA)
V HEALTH FACTORS	VHF(DAS,.DATA)
V IMMUNIZATION	VIMM(DAS,.DATA)
V PATIENT ED	VPEDU(DAS,.DATA)
V POV	VPOV(DAS,.DATA)
V SKIN TEST	VSKIN(DAS,.DATA)

Problem List

The structure of the index is:

```
^PXRMINDX(9000011,CODING SYSTEM,”ISPP”,CODE,STATUS,PRIORITY,DFN,DLM,DAS)
^PXRMINDX(9000011,CODING SYSTEM,”PSPI”,DFN,STATUS,PRIORITY,CODE,DLM,DAS)
```

where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. STATUS is the status of the problem, either active (“A”) or inactive (“I”). PRIORITY can be acute (“A”), chronic (“C”), or null, in which case a “U” is stored. DLM is the Date Last Modified. This structure lets you quickly do things like find active problems whose status is acute.

The API to retrieve the associated data is PROBDATA^GMPLPXRM, it is documented in ICR #5881.

Radiology

The index is on file 70, Rad/Nuc Med Patient. The structure is:

```
 ^PXRMINDX(70,”IP”,PROC,DFN,DATE,DAS)
 ^PXRMINDX(70,”PI”,DFN,PROC,DATE,DAS)
```

where PROC is a pointer to the Rad/Nuc Med Procedures file. The API to retrieve the associated data is EN1^RAPXRM(DAS,.DATA).

The API is documented in ICR #3731.

Registration

The structure of the index is:

```
^PXRMINDX(45,CODING SYSTEM,"INP",CODE,NAME,DFN,DATE,DAS)
^PXRMINDX(45,CODING SYSTEM,"PNI",DFN,NAME,CODE,DATE,DAS)
```

where CODING SYSTEM is a three-character abbreviation as defined in the Coding Systems file (#757.03) and CODE is the code, not the pointer. NAME is the name of the field where the code is stored.

ICD codes are stored in a number of fields in PTF so the storage node information (NODE) is included in the Index to allow quick searching and retrieval of specific nodes. The following tables summarize the fields that are indexed and their node names.

ICD Diagnosis Codes

Field Number	Field Name	Name
79	PRINICIPAL DIAGNOSIS	DXLS
80	PRINICIPAL DIAGNOSIS pre 1986	PDX
79.16	SECONDARY DIAGNOSIS 1	D SD1
79.17	SECONDARY DIAGNOSIS 2	D SD2
79.18	SECONDARY DIAGNOSIS 3	D SD3
79.19	SECONDARY DIAGNOSIS 4	D SD4

79.201	SECONDARY DIAGNOSIS 5	D SD5
79.21	SECONDARY DIAGNOSIS 6	D SD6
79.22	SECONDARY DIAGNOSIS 7	D SD7
79.23	SECONDARY DIAGNOSIS 8	D SD8
79.24	SECONDARY DIAGNOSIS 9	D SD9
79.241	SECONDARY DIAGNOSIS 10	D SD10
79.242	SECONDARY DIAGNOSIS 11	D SD11
79.243	SECONDARY DIAGNOSIS 12	D SD12
79.244	SECONDARY DIAGNOSIS 13	D SD13
45.02,5	ICD 1	M ICD1
45.02,6	ICD 2	M ICD2
45.02,7	ICD 3	M ICD3
45.02,8	ICD 4	M ICD4
45.02,9	ICD 5	M ICD5
45.02,11	ICD 6	M ICD6
45.02,12	ICD 7	M ICD7
45.02,13	ICD 8	M ICD8
45.02,14	ICD 9	M ICD9
45.02,15	ICD 10	M ICD10

ICD Operation/Procedure Codes

Field Number	Field Name	Node Name
45.05,4	PROCEDURE CODE 1	P1
45.05,5	PROCEDURE CODE 2	P2
45.05,6	PROCEDURE CODE 3	P3
45.05,7	PROCEDURE CODE 4	P4
45.05,8	PROCEDURE CODE 5	P5
45.01,8	OPERATION CODE 1	S1
45.01,9	OPERATION CODE 2	S2
45.01,10	OPERATION CODE 3	S3
45.01,11	OPERATION CODE 4	S4
45.01,12	OPERATION CODE 5	S5

In summary, there are a total of 25 ICD diagnosis codes, 15 associated with the discharge and 10 associated with movements. The date for the discharge entries is the discharge date. Movement data is stored in subfile #45.02 so there are 10 possible ICD diagnosis codes for each movement. The date is the MOVEMENT DATE field #45.02, 10.

The names DXLS and PDX don’t bear any resemblance to the field names because these fields were renamed from DXLS to PRINCIPAL DIAGNOSIS and PDX to PRINICIPAL DIAGNOSIS pre 1986 after the Index was released.

The ICD procedure codes stored in subfiles #45.05 and #45.01 are indexed. The dates for these are the PROCEDURE DATE field #45.05.01 and SURGERY/PROCEDURE DATE field #45.01.01 respectively.

The ICD procedure codes stored in the 401P node (fields #45.01, #45.02, #45.03, #45.04, and #45.050 are not indexed because they are valid only for admissions prior to 10/01/1987. This is documented in the User Manual – PTF Menu; located in the Admission Discharge Transfer (ADT) section of the VistA Documentation Library.

The ICD diagnosis codes stored in the CPT RECORD DATE/TIME multiple, field #45.06,.04 are not indexed because CPT data is stored in file #46 not file #45.

The API to retrieve the data is PTF^DGPXRM(DAS,.DATA), it is documented in ICR #4457.

Vitals

The index is on file 120.5, GMRV Vital Measurement. The structure is:

```
 ^PXRMINDX(120.5,”IP”,VITAL TYPE,DFN,DATE,DAS)
 ^PXRMINDX(120.5,”PI”,DFN,VITAL TYPE,DATE,DAS)
```

where VITAL TYPE is a pointer to the GMRV Vital Type file #120.51. Entries that are marked as “entered-in-error” are not indexed. The API to retrieve the associated data is EN^GMRVPXRM(.GMRVDATA,DAS).

The API is documented in ICR #3647.

Summary of the detailed index structure given above

Package	Structure	Pointer	API	S
Inpatient Pharmacy	^PXRMINDX(55,”IP”,DRUG,DFN,START,STOP,DAS) ^PXRMINDX(55,”PI”,DFN,DRUG,START,STOPDAS)	DRUG points to Drug file		3836
Lab	^PXRMINDX(63,”IP”,ITEM,DFN,DATE,DAS) ^PXRMINDX(63,”PI”,DFN,ITEM,DATE,DAS) ^PXRMINDX(63,”PDI”,DFN,,DATE,ITEM,DAS)	ITEM is formatted depending on the type of data	LRPXRM^LRPXAPI(.DATA,DAS,ITEM)	4245
Mental Health	^PXRMINDX(601.84,”IP”,INS,DFN,DATE,DAS) ^PXRMINDX(601.84,”PI”,DFN,INS,DATE,DAS) ^PXRMINDX(601.2,”IP”,INS,DFN,DATE,DAS) ^PXRMINDX(601.2,”PI”,DFN,INS,DATE,DAS)	INS pointer to the MH Instrument file	ENDAS71^YTQPXRM6(.DATA,DAS)	5043
Non-VA meds	^PXRMINDX(“55NVA”,”IP”,POI,DFN,START,STOP,DAS) ^PXRMINDX(“55NVA”,”PI”,DFN,POI,START,STOP,DAS)		NVA^PSOPXRM1(DAS,.DATA)	3793
Order Entry	^PXRMINDX(100,”IP”,OI,DFN,DATE,DAS) ^PXRMINDX(100,”PI”,DFN,OI,DATE,DAS)	OI points to the Orderable Items file	EN^ORX8(DA)	871
Outpatient Pharmacy	^PXRMINDX(52,”IP”,DRUG,DFN,START,STOP,DAS) ^PXRMINDX(52,”PI”,DFN,DRUG,START,STOP,DAS)	DRUG is a pointer to Drug file	PSRX^PSOPXRM1(DAS,.DATA).	3793

PCE
V FILES:
 	
V CPT
V EXAM
V HEALTH FACTORS
V IMMUNIZATION
V PATIENT ED
V POV
V SKIN TEST
	Non-coded values:
	^PXRMINDX(FILE NUMBER,“IP”,ITEM,DFN,DATE,DAS)
	^PXRMINDX(FILE NUMBERr,“PI”,DFN,ITEM,DATE,DAS)
	
	Coded values:
	^PXRMINDX(FILE NUMBER,CODING SYSTEM,”IPP”,CODE,TYPE,DFN,DATE,DAS)
	^PXRMINDX(FILE NUMBER,CODING SYSTEM,”PPI”,DFN,TYPE,CODE,DATE,DAS)
	
	V Immunization:
	^PXRMINDX(9000010.11,"CVX","IP",CVX_CODE,DFN,DATE,DAS)
	^PXRMINDX(9000010.11,"CVX","PI",DFN,CVX_CODE,DATE,DAS)
		Item is the .01 of the V file, for example a pointer to the Education Topic file or Health Factor file	PXPXRM
			ENTRY POINT
			VCPT(DAS,.DATA)
			VXAM(DAS,.DATA)
			VHF(DAS,.DATA)
			VIMM(DAS,.DATA)
			VPEDU(DAS,.DATA)
			VPOV(DAS,.DATA)
			VSKIN(DAS,.DATA)
				4250

Problem List	^PXRMINDX(9000011,CODING SYSTEM”ISPP”,CODE,STATUS,PRIORITY,DFN,DLM,DAS) ^PXRMINDX(9000011,CODING SYSTEM,”PSPI”,DFN,STATUS,PRIORITY,CODE,DLM,DAS)	None	PROB^GMPLPXRM	5881
Radiology	^PXRMINDX(70,”IP”,PROC,DFN,DATE,DAS) ^PXRMINDX(70,”PI”,DFN,PROC,DATE,DAS)	PROC points to the Rad/Nuc Med Procedures file	EN1^RAPXRM(DAS,.DATA).	3731
Registration	^PXRMINDX(45,CODING SYSTEM,”INP”,CODE,DFN,DATE,NODE,DAS) ^PXRMINDX(45,CODING SYSTEM,”PNI”,DFN,CODE,DATE,NODE,DAS) NODE,DAS)	None.	PTF^DGPXRM(DAS,.DATA)	
Vitals	^PXRMINDX(120.5,”IP”,VITAL	VITAL TYPE	EN^GMRVPXRM(.GMRVDAT	3647

	TYPE,DFN,DATE,DAS) ^PXRMINDX(120.5,”PI”,DFN,VITAL TYPE,DATE,DAS)	points to the GMRV Vital Type file	A,DAS).	

Cross-References

The Index is kept current by using new-style FileMan cross-references that fire whenever data is added, edited, or deleted. A list of the cross-references follows.

NOTE: Some of the packages do direct sets of the data into their globals instead of using FileMan. In those cases, the routines where the data is set or killed have been modified to call the package API that does the set or kill of the Index entry.

LAB

Lab results are stored in the Lab Data file #63. This is a very hierarchical file with a strong dependence on the data dictionary.

The Lab package makes programming calls to update the ^PXRMINDX indexes. Chemistry-type data updates the indexes when results are verified. Anatomic Pathology and Microbiology update indexes when results are reported and/or released. Any changes to existing lab data update the indexes. All indexes are set using SLAB^LRPX and killed using KLAB^LRPX.

Routines

Chemistry-type data updates in a central routine.

LRVER3A Chemistry data are updated on verification and editing of verified data. All transactions go through LRVER3A, which stores the results and sets all the cross-references. This routine calls CHSET^LRPX.

LROC It is very rare but chemistry data may be purged during purging of old orders and accessions. This only happens on data that is corrupted and not reportable. This routine calls CHKILL^LRPX.

All Microbiology and Anatomic Pathology data are updated using the same routine, UPDATE^LRPXRM. Adding, editing, or deleting data invokes this call. Results are compared before and after editing. Any change will update the indexes. This routine is called from several routines:

LRAPDA
LRAPDSR
LRMIEDZ
LRMIEDZ2
LRMISTF1
LRMIV
LRMIV1
LRMIV2

Lab Indexes

^PXRMINDX(63,"PI",DFN,ITEM,DATE,DAS)

This index is used for finding results of tests on a patient.

^PXRMINDX(63,"IP", ITEM,DFN,DATE,DAS)
This index is used for finding patients that have specific lab results.
^PXRMINDX(63,"PDI",DFN,DATE,ITEM,DAS)
This index is only used for Microbiology and Anatomic Pathology and is used for finding results on a patient for a specific time period. Chemistry-type data does not require this because the data are already stored in a similar format in the Lab Data file. Micro and AP data use a compound structure for ITEM (the lab test or other coded result) and the "PDI" index provides a faster path to the results. Also, AP data is broken into four sections: Autopsy, Cytology, Electron Microscopy, and Surgical Pathology. This index collates results by collection date/time regardless of the section; again, making retrieval faster.

File #	File Name	Sub-file #	Sub-file Name	Cross-reference
45	PTF			ACRDSD1 ACRDSD10 ACRDSD11 ACRDSD12 ACRDSD13 ACRDSD14 ACRDSD2 ACRDSD3 ACRDSD4 ACRDSD5 ACRDSD6 ACRDSD7 ACRDSD8 ACRDSD9 ACRDDXLS ACRDPDX
		45.01	401	ACRPS1 ACRPS2 ACRPS3 ACRPS4 ACRPS5
		45.02	501	ACRDM1 ACRDM10 ACRDM2 ACRDM3 ACRDM4 ACRD5 ACRD6 ACRD7 ACRD8 ACRD9
		45.05	601	ACRPP1 ACRPP2 ACRPP3 ACRPP4 ACRPP5

52	Prescription			ACRO
		52.1	Refill	ACRR
		52.2	Partial Date	ACRP
55	Pharmacy Patient	55.01	IV	ACRIV
		55.05	Non-VA meds	ACRNVA
		55.06	Unit Dose	ACRUD
63	Lab Data			None
70	Rad/Nuc Med Patient	70.03	Examinations	ACR
100	Order			None
120.5	GMRV Vital Measurement			ACR
601.2	Psych Instrument Patient	601.22	Date	ACR
9000010.07	V POV			ACR
9000010.11	V IMMUNIZATION			ACR
9000010.12	V SKIN TEST			ACR
9000010.13	V EXAM			ACR
9000010.16	V PATIENT ED			ACR
9000010.18	V CPT			ACR
9000010.23	V HEALTH FACTORS			ACR
9000011	Problem			ACR01
9999999.14	Immunization			ACR

Using FileMan to obtain detailed Cross-Reference descriptions

If you wish a more detailed description of any of these cross-references, there are two different ways to get it – both use FileMan.

Method 1, Inquire Option
Use the Inquire option on the Index file.

```
VA FileMan 22.0

Select OPTION: I   INQUIRE TO FILE ENTRIES

OUTPUT FROM WHAT FILE: INDEX//
Select INDEX: ACR
     1   ACR    120.5  Clinical Reminders cross-reference.
     2   ACR    70  Clinical Reminders index.
     3   ACR    9000010.18  Clinical Reminders index.
     4   ACR    9000010.23  Clinical Reminders index.
     5   ACR    9000010.11  Clinical Reminders index.
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-5:
     6   ACR    9000010.16  Clinical Reminders index.
     7   ACR    9000010.07  Clinical Reminders index.
     8   ACR    9000010.12  Clinical Reminders index.
     9   ACR    9000010.13  Clinical Reminders index.
     10  ACR    601.2  Clinical Reminders cross-reference.
Press <RETURN> to see more, '^' to exit this list, OR
CHOOSE 1-10: 6  9000010.16  ACR  Clinical Reminders index.
ANOTHER ONE:
STANDARD CAPTIONED OUTPUT? Yes//   (Yes)
```

```
Include COMPUTED fields:  (N/Y/R/B): NO// BOTH Computed Fields and Record
Number
 (IEN)

NUMBER: 279                             FILE: 9000010.16
  NAME: ACR
  SHORT DESCRIPTION: Clinical Reminders index.
  TYPE: MUMPS                           EXECUTION: RECORD
  ACTIVITY: IR                          ROOT TYPE: INDEX FILE
  ROOT FILE: 9000010.16                 USE: ACTION
 DESCRIPTION:   This cross-reference builds two indexes. One for finding all
 patients with a particular education topic and one for finding all the
 education topics a patient has.  The indexes are stored in the Clinical
 Reminders index global as:
  ^PXRMINDX(9000010.16,"IP",EDUCATION TOPIC,DFN,VISIT DATE,DAS) and
  ^PXRMINDX(9000010.16,"PI",DFN,EDUCATION TOPIC,VISIT DATE,DAS) respectively.
  SET LOGIC: D SVFILE^PXPXRM(9000010.16,.X,.DA)
  KILL LOGIC: D KVFILE^PXPXRM(9000010.16,.X,.DA)
  KILL ENTIRE INDEX CODE: K ^PXRMINDX(9000010.16)
ORDER NUMBER: 1                         TYPE OF VALUE: FIELD
  FILE: 9000010.16                      FIELD: .01
  SUBSCRIPT NUMBER: 1                   COLLATION: forwards
ORDER NUMBER: 2                         TYPE OF VALUE: FIELD
  FILE: 9000010.16                      FIELD: .02
  SUBSCRIPT NUMBER: 2                   COLLATION: forwards
ORDER NUMBER: 3                         TYPE OF VALUE: FIELD
```

Method 2, Data Dictionary Utility

The other way to obtain detailed cross-reference descriptions is to use the Data Dictionary Utility. You can look at an entire file or a sub-file. In the example below, we look at a sub-file.

```
VA FileMan 22.0

Select OPTION:    DATA DICTIONARY UTILITIES
Select DATA DICTIONARY UTILITY OPTION:    LIST FILE ATTRIBUTES
 START WITH WHAT FILE: PTF//
      GO TO WHAT FILE: PTF//
      Select SUB-FILE: 601
Select LISTING FORMAT: STANDARD// INDEXES ONLY
What type of cross-reference (Traditional or New)? Both// NEW
Which field: ALL//
DEVICE:   ANYWHERE    Right Margin: 80//
NEW-STYLE INDEX LIST -- FILE #45                           08/18/04    PAGE 1
-------------------------------------------------------------------------------
```

Subfile #45.05

New-Style Indexes:

```
  ACRPP1 (#1287)    RECORD    MUMPS    IR    ACTION    WHOLE FILE (#45)
      Short Descr:  Clinical Reminders Index for ICD procedure code lookup.
      Description:  This cross-reference builds two indexes, one for finding
                    all patients with a particular ICD procedure code and one
                    for finding all the ICD procedure codes a patient has.  The
                    indexes are stored in the Clinical Reminders Index global
                    as:
                     ^PXRMINDX(45,CODESYS,"INP",CODE,NODE,DFN,DATE,DAS) and
                     ^PXRMINDX(45,CODESYS,"PNI",DFN,NODE,CODE,DATE,DAS)
                    respectively.  CODESYS is the standard three-character
                    abbreviation for the coding system.  DATE is the
                    surgery/procedure date.  NODE is P (for procedure) followed
                    by procedure code number. For example, P1 means it was
                    found on the P node and it was Procedure Code 1.  For
                    complete details, see the Clinical Reminders Index
                    Technical Guide/Programmer's Manual.
        Set Logic:  D SPTFP^DGPTDDCR(.X,.DA,"P",1)
       Kill Logic:  D KPTFP^DGPTDDCR(.X,.DA,"P",1)
             X(1):  PROCEDURE DATE  (45.05,.01)  (Subscr 1)  (forwards)
             X(2):  PROCEDURE CODE 1  (45.05,4)  (Subscr 2)  (forwards)
```

The rest of the cross-references on this sub-file have been left out for brevity.

If you know the field number or field name of a field used in the cross-reference, you can select a single cross-reference for display.

```
VA FileMan 22.0

Select OPTION:    DATA DICTIONARY UTILITIES
Select DATA DICTIONARY UTILITY OPTION:    LIST FILE ATTRIBUTES
 START WITH WHAT FILE: PTF//
      GO TO WHAT FILE: PTF//
      Select SUB-FILE:
Select LISTING FORMAT: STANDARD// INDEXES ONLY
What type of cross-reference (Traditional or New)? Both// NEW
Which field: ALL//    SECONDARY DIAGNOSIS 1
DEVICE:   ANYWHERE    Right Margin: 80//
NEW-STYLE INDEX LIST -- FILE #45, FIELD #79.16              08/18/04    PAGE 1
-------------------------------------------------------------------------------

  ACRDSD1 (#1263)    RECORD    MUMPS    IR    ACTION
      Short Descr:  Clinical Reminders Index for ICD diagnosis code lookup.
      Description:  This cross-reference builds two indexes, one for finding
                    all patients with a particular ICD diagnosis code and one
                    for finding all the ICD diagnosis codes a patient has.  The
                    indexes are stored in the Clinical Reminders Index global
                    as:
                     ^PXRMINDX(45,CODESYS,"INP",CODE,NAME,DFN,DATE,DAS) and
                     ^PXRMINDX(45,CODESYS,"PNI",DFN,NAME,CODE,DATE,DAS)
                    respectively.  CODESYS is the standard three character
                    abbreviation for the coding system.  DATE is the discharge
                    date. If it does not exist then the admission date is used.

                    NAME is the name of the field where the code is stored. An
                    example is D SD1, where D SD signifies it is a discharge
                    secondary diagnosis.  If the TYPE OF RECORD is CENSUS then
                    the entry is not indexed.  For complete details, see the
                    Clinical Reminders Index Technical Guide/Programmer's
                    Manual.
        Set Logic:  D SPTFDD^DGPTDDCR(.X,.DA,"D SD1")
       Kill Logic:  D KPTFDD^DGPTDDCR(.X,.DA,"D SD1")
             X(1):  PATIENT  (45,.01)  (Subscr 1)  (forwards)
             X(2):  ADMISSION DATE  (45,2)  (Subscr 2)  (forwards)
             X(3):  TYPE OF RECORD  (45,11)  (Subscr 3)  (forwards)
             X(4):  SECONDARY DIAGNOSIS 1  (45,79.16)  (Subscr 4)  (forwards)
             X(5):  DISCHARGE DATE  (45,70)
```