Complicated Searches in VistA

Joel L. Ivey, Ph.D.

joelivey@gmail.com

jivey@jiveysoft.com

A Product of 27 years – Some History

- This work started as a means to see how relationships in VistA (then DHCP) data could be viewed as if it were in an SQL database, since I heard about SQL databases.
- Across time I turned back to the relationships in VistA and worked to identify how to access that data. Part of this was building a table of the far relationships, etc., which we later learned was basically a metadata dictionary of the FileMan database.
- This work turned into a method to perform searches based on data across the entire FileMan database.
- This work was demonstrated for other Kernel team members, supervisors (Catherine Pfeil, Maureen Hoye) and management (Dan Bishop), they were all impressed, but never gave permission to release it.
- Basically, the final work on it was in 2000, until recently moving it to my own namespace and getting it ready to release to the Open Source community.

The Metadata Dictionary

- If a table of relationships hasn't been built recently, this table is built for a specific file.

- VISTA>D ^KBBPYSHO
- Press Enter NOW to run the application or wait 5 seconds to run the Unit Tests:
- First I have to rebuild the field cross-reference file
- THIS MAY TAKE A WHILE <- - - - Building the file
- Select File:

- VISTA>D ^%G
- Global ^XTMP("KBBPY",)
- ^XTMP("KBBPY",0)="3170819^3170812^KBBPY METADATA DICTIONARY OF FileMan Files"
- "FNM")=3170812.143
- "XR1")pointer

The FNM and XR1 nodes of ^XTMP("KBBPY"

FNF – file names and sub file names

^XTMP("KBBPY","FNM","*ADDITIONAL PLANS",99.18)="99.18;99.17;99.16;99^TPA;1;1"

subfile_name subfile# subfile3;upfile2;upfile1;topfile^node1; node2;node3

^XTMP("KBBPY","FNM","*ADDITIONAL PRINT LOCATIONS",713.025)="713.025;713^2"

^XTMP("KBBPY","FNM","*AMIS/RCS 14-4",60.06)="60.06;60.01;60^1;2"

^XTMP("KBBPY","FNM","*ASSESSMENT",2040.012)="2040.012;2040^11“

^XTMP("KBBPY","FNM","$STACK()",3.7519)="3.7519;3.0751;3.075^1;STACK“

XR1 – cross reference of fields

^XTMP("KBBPY","XR1","# ACTIVE IN BATCH",791812,791812,.04)="0;4^^791812^"

^XTMP("KBBPY","XR1","# COLLATERALS",115.7,115.701,2) = "0;3^ P^ 115.701;115.7^"

field_name primary,subfile,field# node;piece^ node for subfile^ subfiles; ; topfile^XREF

^XTMP("KBBPY","XR1","# DAILY MODIFIED DIETS",117,117,57)="1;26^^117^"

^XTMP("KBBPY","XR1","# DAILY MODIFIED DIETS",117,117.06,32)="0;27^2^117.06;117^"

^XTMP("KBBPY","FNM","PATIENT",2)="2^" <- - - - - Files and Subfiles with name of PATIENT (20)

```
  42.51)="42.51;42.5^P"
 44.003)="44.003;44.001;44^S;1"
 44.007)="44.007;44.006;44^C;1"
 50.801)="50.801;50.8^1"
 50.806)="50.806;50.805;50.803;50.8^2;2;1"
 53.401)="53.401;53.4^1"
53.4102)="53.4102;53.4101;53.41^1;1"
53.43011)="53.43011;53.4301;53.43^1;1"
53.4401)="53.4401;53.44^1"
 53.51)="53.51;53.5^1"
115.701)="115.701;115.7^P"
133.801)="133.801;133.8^1"
 183.21)="183.21;183.2^1"
 213.51)="213.51;213.5^1"
 409.69)="409.69;409.6^1"
 617.01)="617.01;617^1"
665.72319)="665.72319;665.7231;665.723;665.72^1;1;V"
 810.16)="810.16;810.1^2"
2006.552)="2006.552;2006.55^1"
```

Fields With Name of Patient (a few of 215)

^XTMP("KBBPY","XR1","PATIENT",26.21,26.21,.01)="0;1^^26.21^""B"""" <---- Lowest file # (other than 2)

^XTMP("KBBPY","XR1","PATIENT",27.11,27.11,.02)="0;2^^27.11^""C""""

.....

^XTMP("KBBPY","XR1","PATIENT",45.5,45.5,1)="0;2^^45.5^""PT""""

^XTMP("KBBPY","XR1","PATIENT",45.87,45.87,.09)="0;9^^45.87^"

^XTMP("KBBPY","XR1","PATIENT",50.0731,50.0731,7)="0;8^^50.0731^"

^XTMP("KBBPY","XR1","PATIENT",50.8,50.8,1)="1;0^^50.8^"

^XTMP("KBBPY","XR1","PATIENT",50.8,50.801,.01)="0;1^1^50.801;50.8^"

^XTMP("KBBPY","XR1","PATIENT",50.8,50.805,8)="1;0^2;2^50.805;50.803;50.8^"

^XTMP("KBBPY","XR1","PATIENT",50.8,50.806,.01)="0;1^2;2;1^50.806;50.805;50.803;50.8^"

^XTMP("KBBPY","XR1","PATIENT",52,52,2)="0;2^^52^"

^XTMP("KBBPY","XR1","PATIENT",52.11,52.11,.01)="0;1^^52.11^""B""""

^XTMP("KBBPY","XR1","PATIENT",52.41,52.41,1)="0;2^^52.41^""P""""

KBBPYSHO – A Tour of Relationships - 1

Select File: PATIENT

Select From:

1. PATIENT (2)
2. PATIENT (42.51)
3. PATIENT (44.003)
4. PATIENT (44.007)
5. PATIENT (50.801)
6. PATIENT (50.806)
7. PATIENT (53.401)
8. PATIENT (53.4102)
9. PATIENT (53.43011)
10. PATIENT (53.4401)

Enter '^' to STOP or

Select 1 to 10:

Select File: 2

Select From:

1. PATIENT (2)
2. ENROLLMENT CLINIC (2.001)
3. ALIAS (2.01)
4. ENROLLMENT DATA (2.011)
5. RACE INFORMATION (2.02)
6. PATIENT ELIGIBILITIES (2.0361)
7. RATED DISABILITIES (VA) (2.04)
8. SERVICE CONNECTED CONDITIONS (2.05)
9. PH DATE/TIME UPDATED (2.0534)
10. ETHNICITY INFORMATION (2.06)

Enter '^' to STOP or

Select 1 to 10:

KBBPYSHO - 2 -The PATIENT File Itself

B KEY - PATIENT(2) - NAME .... (Free Text, 3 to 30 chars)

*AMOUNT OF MILITARY RETIREMENT .... (Numeric, 10 digits, 2 decimals places)

*AMOUNT OF SOCIAL SECURITY .... (Numeric, 10 digits, 2 decimals places)

*CLAIM FOLDER LOCATION .... (Free Text, 2 to 40 chars)

*CURRENT PC PRACTITIONER .... (pointer to entry in NEW PERSON file)

*CURRENT PC TEAM .... (pointer to entry in TEAM file)

*1U4N .... (Computed)

*2ND MOST RECENT DATE OF CARE .... (Date)

*2ND MOST RECENT LOCATION .... (pointer to entry in INSTITUTION file)

A&A AMOUNT .... (Numeric, 10 digits, 2 decimals places)

ABSENCE DIVISION .... (Computed)

ADDRESS CHANGE DT/TM .... (Date)

ADDRESS CHANGE SITE .... (pointer to entry in INSTITUTION file)

ADDRESS CHANGE SOURCE .... (Set of Codes)

ADDRESS CHANGE USER .... (pointer to entry in NEW PERSON file)

AGE .... (Computed)

Enter '^' to STOP or <ret> to continue... ^ <- - - - skip to related files

KBBPYSHO - 3 –Subfiles of the PATIENT file

TABLES RELATED TO PATIENT FILE (2)

In the M world, these tables are subfiles to the PATIENT file.

<There are 32 subfile tables for the 'PATIENT' file>

PATIENT - ENROLLMENT CLINIC (2.001)

PATIENT - ENROLLMENT CLINIC - ENROLLMENT DATA (2.011)

PATIENT - ALIAS (2.01)

PATIENT - RACE INFORMATION (2.02)

PATIENT - PATIENT ELIGIBILITIES (2.0361)

PATIENT - RATED DISABILITIES (VA) (2.04)

PATIENT - SERVICE CONNECTED CONDITIONS (2.05)

PATIENT - PH DATE/TIME UPDATED (2.0534)

PATIENT - ETHNICITY INFORMATION (2.06)

PATIENT - ICN HISTORY (2.0992)

PATIENT - CMOR HISTORY (2.0993)

Enter '^' to STOP or <ret> to continue... ^ <- - - - skip to 'pointed to' related files

KBBPYSHO - 4 – Files related by pointers

TABLES RELATED TO PATIENT FILE (2)

In the M world, the following tables are files and subfiles that contain links

to the PATIENT file

(an * indicates that the link is the primary identifier (.01) for the file,

an *X indicates that the link is also DINUMed to the PATIENT file

<There are 316 related tables for the 'PATIENT' file>

PRF HL7 EVENT (26.21)*

PATIENT ENROLLMENT (27.11)

ENROLLMENT QUERY LOG (27.12)*

ENROLLMENT/ELIGIBILITY UPLOAD AUDIT (27.14)

NOSE AND THROAT RADIUM HISTORY (28.11)*

MST HISTORY (29.11)

DG SECURITY LOG (38.1)*X

INCONSISTENT DATA (38.5)*X

Enter "^" to STOP or <ret> to continue... ^

KBBPYQRY – Handling Complicated Searches

- VISTA>D ^KBBPYQRY

- Press Enter NOW to run the application or wait 5 seconds to run the Unit Tests:
- Select Primary File: 2 PATIENT
- Is this the correct file? YES//
- .....
- .....|.....
- .....|.....
- .....
- .....|.....
- .....|.....
- .....|.
- .....
- .....|.....
- .....|.....
- .....|....
- .....|....
- 1 second

KBBPYQRY – The First Field And Condition

Set of fields and conditions for Test 1

/*

The user is prompted for fields and associated conditions, since at this level, the program is going to be selecting individuals based on the specified criteria to be included in the output. There may be multiple sets of fields and conditions included in the analysis, this is only the first.

The field must be specified by at least part of the field name (field numbers will not currently work), and optionally the file number for the field (this may result in many, many choices - you probably don't want to enter NAME or DATE without a file number).

Then a prompt for the condition on X - this is required since this is part of the selection based on criteria.

*/

Field name: SEX@2

1 SEX in 'PATIENT' file

Code for condition on X: X="M"

KBBPYQRY – Selecting only 1 field for this condition

Multiple fields with associated conditions may be entered under one set. If more than one field and condition specification is entered, they will all be considered as ANDed together, so that all of the specified conditions must be met for the entry to be selected.

Since the conditions being specified for this example will be ANDed together this could have been done, but instead a second set of conditions will be used.

Field name:

KBBPYQRY – The next set of fields and Conditions

/*

The following prompts for input for a second set of fields and conditions.

Simply entering return would indicate that another set of fields and is not desired. However, a second set will be entered here.

*/

Set of fields and conditions for Test 2

Field name: GENERIC

1 GENERIC NAME in 'DRUG' file

2 GENERIC PROVIDER in 'PRESCRIPTION' file

3 GENERIC PROVIDER in 'REFILL' sub-file of 'PRESCRIPTION' file

4 GENERIC PROVIDER in 'PARTIAL DATE' sub-file of 'PRESCRIPTION' file

5 GENERIC EQUIVALENT PRODUCT ID in 'TRANSACTIONS' sub-file of 'BPS CLAIMS' file

6 GENERIC EQVLNT PRODUCT ID QLFR in 'TRANSACTIONS' sub-file of 'BPS CLAIMS' file

Enter '^' to exit OR

Select (1 to 6 [out of 6 entries]) : 1

KBBPYQRY – Oops - Lots of ways to get there

/*

The file and field selected for analysis is related to the primary file, but it can be reached in multiple ways, so the user is asked to choose the connection pathway from those available. All 24 choices are not shown here, since the third choice was the desired pathway, but a user could view all of them before making a choice.

Once a choice of paths between two files is selected, any other fields in files that might be on this pathway would be included without asking for paths, since this one is already selected.

*/

Select Method of Connection Between Files (24 choices) from:

1 a. NAME field of the PATIENT file is POINTED TO by the PHARMACY PATIENT file (55)

b. DISPENSE DRUG field of the PHARMACY PATIENT sub-file (#55.05) of the PHARMACY PATIENT file POINTS to the DRUG file (50)

2 a. NAME field of the PATIENT file is POINTED TO by the PHARMACY PATIENT file (55)

b. *SOLUTION field of the PHARMACY PATIENT sub-file (#55.06) of the PHARMACY PATIENT file POINTS to the DRUG file (50)

3 a. NAME field of the PATIENT file is POINTED TO by the PHARMACY PATIENT file (55)

b. PRESCRIPTION PROFILE field of the PHARMACY PATIENT sub-file (#55.03) of the PHARMACY PATIENT file POINTS to the PRESCRIPTION file (52)

c. DRUG field of the PRESCRIPTION file POINTS to the DRUG file (50)

Select Method of Connection (1 to 3 out of 24): 3

Code for condition on X: X["ASPRIN"

KBBPYQRY – Combining Sets via AND or OR

// Again, no response for another field and file in this set of conditions

Field name:

Set of fields and conditions for Test 3

// And no response here, since only two conditions were to be specified

Field name:

/*

Since there were multiple sets of conditions specified, it is necessary for the user to indicate which of these sets should be ANDed to each other. It is possible to enter sets of conditions to be ANDed together, while another set might be ANDed together. The remaining sets that are not included in an AND set would be ORed together with the other sets.

In the following sets 1 and 2 are ANDed together.

*/

The response(s) must be a comma-separated series of digits ranging from 1 to 2.

Used values from one response may be used in a subsequent response as necessary

Enter numbers for TESTS to be 'AND'ed together (enter ?? for help): : (1-2): 1,2//

KBBPYQRY – Selecting data for output

/*

The user is then requested to specify what fields should be output for the entries that meet the criteria specified. As it indicates, for the output of data from the selected criteria, conditions are optional, but may be very desirable (e.g., if conditions are not specified for the Generic Name in the Drug file, then all Generic Names that the selected user has been prescribed via the selected pathway would be included in the output).

The fields selected here are the Patient's Name, the sex, and Generic Name for the Drug prescribed. Since SEX is a single entry field, it really doesn't require a condition at this point for the selected patients.

If multiple fields are specified with conditions, they may be ANDed or ORed.

*/

Data to **Report** for Matches (Conditions are optional)

Field name: NAME@2

1 NAME in 'PATIENT' file

2 NAME in 'ELIGIBILITY/BENEFIT' sub-file of 'INSURANCE TYPE' sub-file of 'PATIENT' file

3 NAME in 'CONTACT INFORMATION' sub-file of 'ELIGIBILITY/BENEFIT' sub-file of 'INSURANCE TYPE' sub-file of 'PATIENT' file

4 NAME COMPONENTS in 'PATIENT' file

5 NAME OF INSURED in 'INSURANCE TYPE' sub-file of 'PATIENT' file

Enter '^' to exit OR

Select (1 to 5 [out of 5 entries]) : 1

Code for condition on X:

KBBPY – Selecting other output fields

Field name: SEX@2

1 SEX in 'PATIENT' file

Code for condition on X:

Field name: GENERIC

1 GENERIC NAME in 'DRUG' file

2 GENERIC PROVIDER in 'PRESCRIPTION' file

3 GENERIC PROVIDER in 'REFILL' sub-file of 'PRESCRIPTION' file

4 GENERIC PROVIDER in 'PARTIAL DATE' sub-file of 'PRESCRIPTION' file

5 GENERIC EQUIVALENT PRODUCT ID in 'TRANSACTIONS' sub-file of 'BPS CLAIMS' file

6 GENERIC EQVLNT PRODUCT ID QLFR in 'TRANSACTIONS' sub-file of 'BPS CLAIMS' file

Enter '^' to exit OR

Select (1 to 6 [out of 6 entries]) : 1

Code for condition on X: X["ASPRIN"

KBBPYQRY – No Further Output Selections

Field name:

PATIENT NAME is not a multiple and will be included on each line of output

PATIENT SEX is not a multiple and will be included on each line of output

Select which of the following should be 'AND'ed together, entry of only one number will 'OR' that value with the others. Entry of 1,2 would have 1 and 2 ANDed together, while 1,3 would also have 1 and 3 ANDed together.

/*

* Really this shouldn't be necessary for a single condition

*/

1 DRUG GENERIC NAME *** X["ASPRIN"

Response(s) must be a comma-separated series of digits ranging from 1 to 1.

Used values from one response may be used in a subsequent response as necessary

Enter numbers for TESTS to be 'AND'ed together (enter ?? for help): : (1-1): 1//

KBBPYQRY – Selection of output format

- /*
- * The user is given multiple possible output formats, Unpacked (basically
- * a regular text output) or Packed with separators between the fields
- * (either '^'-delimited, or ','-delimited for Excel). The results for
- * this search will be shown in both of these packed formats as well.
- */
- The output may be PACKED with only a separator between data fields,
- or the output may be in a more readable format
- Output Packed? NO

KBBPYQRY – The Output Routine

/*

The specification for the analysis are added to the end of a template routine (KBBPYROU) which contains all of the code for performing the searches and analyses. The specifications added to the end provide all of the necessary information to perform the search and output and may be used in the future to perform the analysis again. Another option would be to place the specifications into a file so that it could be performed again as desired.

The template routine is saved to the active routine using the KBBPZ namespace followed by 3 digits, beginning with 001 up to 998 (KBBPZ999 is reserved for running unit tests). Before asking the user for input, the program checks to see if the high value has already been used, and if so the user is informed that the routine can't be generated and the analysis can't be performed until some routines are deleted.

*/

The routine to perform the specified analysis has been saved as

KBBPZ024

You may use this routine name in the future to repeat the analysis

KBBPYQRY – The Output 1

DEVICE: HOME// Console (Cache' on Windows)

Selection based on

PATIENT SEX (file #2) *** I X="M"

and DRUG GENERIC NAME (file #50) *** I X["ASPRIN"

DATA VALUES DISPLAYED ARE:

PATIENT NAME

PATIENT SEX

DRUG GENERIC NAME *** condition: I X["ASPRIN"

TEST,PATIENTAAA MALE ASPRIN

TEST,PATIENTAAA MALE ASPRIN_80

KBBPYQRY – The Output 2

/*

The following shows the output from a similar analysis in which the ','-delimited format (for use with Excel) was specified

*/

"TEST,PATIENTAAA","MALE","ASPRIN",,

"TEST,PATIENTAAA","MALE","ASPRIN_80",,

/*

The following shows the output from a similar analysis in which the '^'-delimited format was specified

*/

TEST,PATIENTAAA^MALE^ASPRIN^

TEST,PATIENTAAA^MALE^ASPRIN_80^

Summary

- The KBBPYSHO and related routines provide the capability to see relationships between VistA files and subfiles and other files and/or subfiles in a table form which shows lookup cross references as well as the type of data in the field.
- The KBBPYQRY and related routines provide the capability to perform searches based on multiple fields and conditions which would be difficult to perform with FileMan searches or other means.
- The KBBPY* routines have 97% coverage with their unit tests.
- These files will be provided to OSEHRA for certification
- And are currently available from https://github.com/joelivey/KBBPY-VistA-Show-and-Query

Questions??

Ask now or contact

Joel Ivey

joelivey@gmail.com

jivey@jiveysoft.com