Crossing Applications without using Fragments

Presented By:
Joe Cocuzzo
Sr VP, Reporting Services
MUSE 2015 Workshop

Fragments

- Advantages:
  - Build on your RW knowledge
    - If you can write report, you can write fragment
  - Can cross to any application
  - Can do multi-record selection reports

Fragments

- Disadvantages
  - S..L..O..W..
  - Two (or more) reports to maintain

Demonstration

- Fragment ABS to BAR

```
RUN DATE: 10/17/06              Hallmark Health Abstracting *TEST*
  PAGE 7
RUN TIME: 0738                   Cross to Billing using Fragment
RUN USER: IATRIC
Name                            Balance
Z2NURS,THREE                     21054.00
Z2NURS,FOUR                      21054.00
Z2NURS,BABY                      21070.00
Z2NURPC,WENDI                   114957.00
Z2NURPC,MARY                    110004.00
Z2REHAB,PLAY1                    73656.00

Elapsed Time:        55  SECONDS  for   342 RECORDS
```

Demonstration - continued

- Open prefixes yourself, per each record

```
RUN DATE: 10/17/06              Hallmark Health Abstracting *TEST*
  PAGE 7
RUN TIME: 0741        Cross to Billing No Fragment / Prefix open per record
RUN USER: IATRIC
Name                          Balance
ZZNURS,THREE                   21054.00
ZZNURS,FOUR                    21054.00
ZZNURS,BABY                    21070.00
ZZNURPC,WENDI                 114957.00
ZZNURPC,MARY                  110004.00
ZZREHAB,PLAY1                  73656.00

Elapsed Time:         3  SECONDS for   342 RECORDS
```

Demonstration – Open prefix at start to BAR, restore at end of report.

```
RUN DATE: 10/17/06              Hallmark Health Abstracting *TEST*
  PAGE 7
RUN TIME: 0742              Cross to Billing Open Prefix 1x per Report
RUN USER: IATRIC
Name                          Balance
ZZNURS,THREE                   21054.00
ZZNURS,FOUR                    21054.00
ZZNURS,BABY                    21070.00
ZZNURPC,WENDI                 114957.00
ZZNURPC,MARY                  110004.00
ZZREHAB,PLAY1                  73656.00

Elapsed Time:         0  SECONDS for   342 RECORDS
```

Eliminating Fragments

- Much faster execution time.
- All code in 1 report

- Requires knowledge of prefixes
- Requires ability to write loops

Goals for Workshop

- Understand prefixes for magic
- How to open data and dictionary files in any application

------- MAGIC -------

  – $["SEGS"]
  – %Z.link()

    ------CLIENT SERVER----

    %Z.link.db()

- Review DO and @Next() syntax

Magic Prefixes

Prefix	Used for
#	Input from keyboard / output to screen
!	Output to printer
$	System data and system programs
%	Data files in the current directory Any programs in the current directory
*	Application data
:	Application data
?	Application data
&	Application dictionary
\	MIS dictionaries
/	Memory based temporary file

Prefixes

- # - output to screen, input from terminal

Example code from Report Screen program

```
ABS.PAT.zcus.is.toronto.frag.example.S,AA,HHS.TEST.5.4.MIS,5.4,06.10.17.0700,"COLOR",
>$.[/.T]>T,
1>/ZSPOOL.NEW.JOB,
1>/Z.SCHED.RPT,
IF{/.GUI 1>/G.MSE,
        %ABS.PAT.zcus.is.toronto.frag.example.S["M"]("ST"),
        %ABS.PAT.zcus.is.toronto.frag.example.S["M"]("INIT")},
IF{/.BUFF D(#B)},
IF{/exit.nub ""></exit.nub;D(2,0)>#},
D(23,6,4-/.P,36,1)>#,
36>/.WW,
4-/.P>/.WH,
IF{\GU[/.USR]BPS V(\GU[/.USR]BPS)>#;%["Z.chg.palettes">E] %[E]("SCRM")},
IF{/.GUI D(30)_"CC"_{("Cross to Billing using Fragment":36TL,
                    "")}>#,
```

Prefixes

- !, output to printer

```
%Z.rw.fragment("BAR.PAT.zcus.is.toronto.frag.example.fr.R",DB),
/R.FRAG.VAL>BAL,
IF{/R.FOUND IF{/.LL<1 %[/R.NEW.PAGE.PGM](0)},
/.LL-1>/.LL,
%Z.rw.index(1,"PT STATUS",djd,"FREE"),
(BAL)>/V[2],
I((X¦1:30TL)_" "_IF{/V[2]>X1 X1:2D:9R;" ":9})N>/R.OUT,
IF{%[/R.TRANS]'=% %[/R.TRANS](0);/R.OUT>!}}}}}},
IF{/R.FRAGMENT;
```

Prefixes

- $ - System utility programs, system data
  - $SEG, $DIR, $FE, $WHO.D, $WHO.Q
  - $L.TO.U $U.TO.L
- % - Directory (application) programs, files.
  - %Z programs, application programs
  - application data and dictionary files

Examples from Magic System

```
AA:HHS.TEST.5.4.MIS (NPR)} $WHOD(0)
```

```
Job List For Segment: AA   Directory: HHS.TEST.5.4.MIS      06/10/15 - 15:26
-------------------------------------------------------------------------------
Job   Terminal   M/S Priority   Signon Time  Program
-------------------------------------------------------------------------------

 19              M   4 10 4    04/23 02:12  NMI.OUX.bkg
284              M   4 10 4    08/30 15:19  MIS - REPORT SCHEDULER
326              M   4 10 4    09/29 09:38  MRI BKG MRI.HHS
366              M   4 10 4    05/19 08:55  Passport SCH BKG Job
370     13N15    M   4 10 4    10/15 10:14  APPL DB: NPR
493     13N13    M   4 10 4    10/15 15:25  FEC
559              M   4 10 4    05/16 11:38  Passport File Maintain BKG Job
815              M   4 10 4    05/24 11:15  ADM BKG ADM.HHS
1806     13N9    M   4 10 4    10/15 12:00  APPL DB: NPR

AA:HHS.TEST.5.4.MIS (NPR)}
```

Examples in %

```
AA:HHS.TEST.5.4.MIS (NPR)} %Z.date.in.magic(S(0))
20061015
AA:HHS.TEST.5.4.MIS (NPR)}
```

Client Server Prefixes

- Unique per application
- No need to know where data file lives
- No need to “restore” prefixes, unless crossing universes
- No need to close prefixes at end of report

```
location                  10 FREE    L  MIS.LOCN        $(A)AA[aa]|12
maiden.other.name         30 NAME    L                 $(A)AA[aa]B|4
marital.status            10 FREE    L  MIS.MAR.STATUS  $(A)AA[aa]A|4
med.nec.admit.dx          10 FREE    L                 $(A)AA[aa]O3|15
med.nec.admit.dx.conf      1 YN      L                 $(A)AA[aa]O3|16
med.nec.admit.dx.version  10                           $(A)AA[aa]O3|19
medications               50                           $(A)AA[aa]T|3
mothers.first.name        20                           $(A)AA[aa]B|3
mothers.urn               30                           $(A)AA[aa]O3|9
mri.drc.patient           30 FREE    L
mri.urn                   25 URN     L  MRI.PAT         $(A)AA[aa]|4
name                      30 FREE    L                 $(A)AA[aa]|1
name.w.vip.flag           30 FREE    L                 VAL=(IF($ADM.PAT.vip="Y" @ADM.PARAM.vip.flag)_@ADM.PAT
                                                        VAL=.name)
```

Prefix $(A) instead of *

Client Server

No reason to use fragments in C/S if you can write loops and link data.

Open all the applications you need at start of report.

Opening database also allows you to call ALL PROGRAMS in the database.

C/S Example:

```
OPEN.DBS
@.db#"1."→DB,
%Z.link.db({"EDM."_DB},{"O","B",1}),
%Z.link.db({"RXM."_DB},{"O","B",1}),
%Z.link.db({"NUR."_DB},{"O","B",1}),
%Z.link.db({"LAB."_DB},{"O","B",1}),
%Z.link.db({"PHA."_DB},{"O","B",1}),
%Z.link.db({"OE."_DB},{"O","B",1}),
%Z.link.db({"RAD."_DB},{"O","B",1}),
%Z.link.db({"ADM."_DB},{"O","B",1})
```

MU Stage One Quality Reports "Start" macro

STAR WARS MEMORY TRICK "Obi-Wan" = "O" "B" "1" or {"O","B",1}

Magic: Where is the data?

- `%.APP.data[MIS,APPDB]`

  `%.BAR.data["HHS","BAR.HHS"]`

  `%.BAR.dic["HHS","BAR.HHS"]`

  `MIS dictionaries`

  `%.MIS.dic["HHS","MIS"]`

Example from Front End

- Front end = %FEC(0) programmers interface

```
AA:HHS.TEST.5.4.MIS (NPR)} OPEN ADM
  / = P          Opened to Memory based RAF
  & = 1229839R   Application Dictionary Opened to %.ADM.dic[HHS5.4,ADM.HHS]
  \ = 1401594R   MIS Dictionary Opened to %.MIS.dic[HHS5.4,MIS]
                 Data files opened by ADM.TOOL.BOX.open.prefixes
AA:HHS.TEST.5.4.MIS (NPR)}
```

Data Prefixes

- Example from Report Object Code

```
(S(0)+/.T2D)→START,
IF "IN"'=D(127);D(1)}→djd,
-?D2D[djd]→djd,
DO{+?D2D[djd]→djd'>"IN"&'/R.LIMIT IF{/b.dis.date'=D(127);D(1)}→dzDD,
-?D2D[djd,dzDD]→dzDD,
DO{+?D2D[djd,dzDD]→dzDD'>(/e.dis.date_D(127))&'/R.LIMIT ""→dzR,
DO{+?D2D[djd,dzDD,dzR]→dzR&'/R.LIMIT ?D2D[djd,dzDD,dzR]→dz,
:D2[dz]→X,
```

? for dis.date.x

: for abs.pat data segment

Magic: To get data in other application

- Change $ to the correct segment
- Change % to the correct directory
- Open a prefix to the data file
- Get your data
- Restore prefix to old state
- Go back to original segment
- Go back to original directory

Prefixes - continued

Prefix	Used for
#	Input from keyboard / output to screen
!	Output to printer
$	System data and system programs
%	Data files in the current directory Any programs in the current directory
*	Application data
:	Application data
?	Application data
&	Application dictionary
\	MIS dictionaries
/	Memory based temporary file

SEGS

- $SEGS(SEG,DIR) will move to segment SEG and directory DIR, saving old "location"
  - $ prefix will be "stacked" opened to new segment
  - % prefix will be "stacked" opened to new segment.

Stacked

Demonstration of SEGS

```
AA:HHS.TEST.5.4.MIS (NPR)} $SEGS("HH","HHS.TEST.5.4.MIS")
```

Current Segment and Directory

Destination Seg and Dir

```
AA:HHS.TEST.5.4.MIS (NPR)} $SEGS("HH","HHS.TEST.5.4.MIS")
HHS.TEST.MIS *AA* 5.4 SAT HH
HH:HHS.TEST.5.4.MIS (Sat)}
```

Now we've moved

Demonstration of $SEGS

```
AA:HHS.TEST.5.4.MIS (NPR)} $SEGS("HH","HHS.TEST.5.4.MIS")
HHS.TEST.MIS *AA* 5.4 SAT HH
HH:HHS.TEST.5.4.MIS (Sat)} C($U,%U)
```

Unstack

```
AA:HHS.TEST.5.4.MIS (NPR)} $SEGS("HH","HHS.TEST.5.4.MIS")
HHS.TEST.MIS *AA* 5.4 SAT HH
HH:HHS.TEST.5.4.MIS (Sat)} C($U,%U)

AA:HHS.TEST.5.4.MIS (NPR)}
```

Back to original Seg and Dir

SEGS in a report

- Go to segment and directory of other application

`$SEGS("F","MRI.CVH")`

- Open and stack prefix to application data

`O(?S,%.MRI.data["CVH","MRI.CVH"])`

- Get data needed:

`@MRI.DRC.pat.city^CITY`

- Restore prefixes

`C($U,%U,?U)`

To fool syntax checker

- Go to segment and directory of other application

`$["SEGS"]("F","MRI.CVH")`

- Open and stack prefix to application data

`ZZ%O(?S,%.MRI.data["CVH","MRI.CVH"])`

- Get data needed:

`@MRI.DRC.pat.city^CITY`

- Restore prefixes

`C($U,%U,?U)`

To avoid hard-coding SEG and DIR

```
@mri.urn^MRI.PAT.medical.record,
"MRI."_(@.db#"1.")^DB,
DB^MIS.APPL.database,
$["SEGS"](@MIS.APPL.database.segment,@MIS.APPL.database.directory),
ZZ%OP(?S,%.MRI.data[@.mis,DB]),
@MRI.DRC.pat.street^STREET,
@MRI.DRC.pat.street2^STREET2,
@MRI.DRC.pat.city^CITY,
@MRI.DRC.pat.state^STATE,
@MRI.DRC.pat.zip^ZIP,
C(?U,$U,%U)
```

ABS to MRI for address

```
Enter/Edit Macro Logic: ABS.PAT.zcus.is.toronto.segs.each.record.M.deta
@mri.urn>MRI.PAT.medical.record,
"MRI."_(@.db#"1.")>DB,
DB>MIS.APPL.database,
$["SEGS"](@MIS.APPL.database.segment,@MIS.APPL.database.directory),
ZZ%OP(?S,%.MRI.data[@.mis,DB]),
@MRI.DRC.pat.street>STREET,
@MRI.DRC.pat.street2>STREET2,
@MRI.DRC.pat.city>CITY,
@MRI.DRC.pat.state>STATE,
@MRI.DRC.pat.zip>ZIP,
C(?U,$U,%U)
```

```
Edit Footnotes
D 44
AL D detail
```

Sample Output

```
"ZZNURS,FOUR","585 LEBANON STREET","","MELROSE","MA","02176"
"ZZS1,ONE","","","","","",""
"ZZREHAB,PLAY1","1 MAIN ST","","MELROSE","MA","02176"
"ZZREHAB,PLAY2","1 MAIN ST","","MELROSE","MA","02176"
"ZZADM,CONNOR JAMES","101 MAIN STREET","","MEDFORD","MA","02155"
"ZZCOLD,REVIEW","","","","","",""
"ZZCOLD,TIMOTHY","20 PORTER STREET","","MELROSE","MA","02176"
"ZZNURS,ONE","585 LEBANON ST","","MELROSE","MA","02176"
"ZZNURS,TWO","585 LEBANON ST","","MELROSE","MA","02176"
"ZZNURS,THREE","585 LEBANON ST","","MELROSE","MA","02176"
"ZZNURS,FOUR","585 LEBANON STREET","","MELROSE","MA","02176"
"ZZNURS,BABY","585 LEBANO STREET","","MELROSE","MA","02176"
"ZZNURPC,WENDI","66 DEVILS WAY","","DRACUT","MA","01826"
"ZZNURPC,MARY","78 WEST STREET","","STONEHAM","MA","02180"
"ZZREHAB,PLAY1","1 MAIN ST","","MELROSE","MA","02176"
                                                    End of r
```

Z.link vs SEGS

- %Z.link
  - Knows “how to find” application, takes care of going to the segment and directory, opening, returning to original directory
- $SEGS
  - Needed if you have to call a program in the other application (computed field)

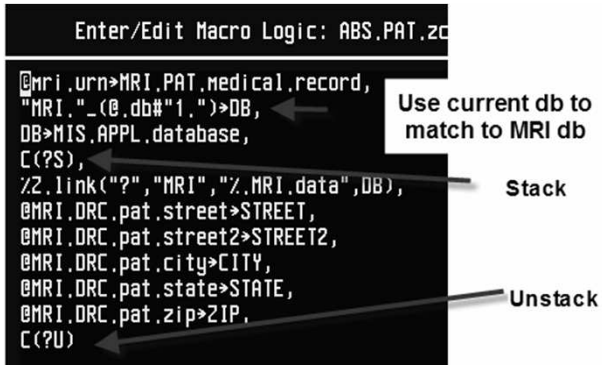
Z.link vs SEGS continued

- Z.link
  - Possible less vulnerable to future syntax checker “enhancements”
- SEGS
  - Needed when opening your own files, non MEDITECH data files (Array, MSM/PICIS, LSS, Iatric Systems)

Example..MOCK LAB

```
A^ACCT,
B^TEST,
"LAB.EDW"^MIS.APPL.database,
$["SEGS"](@MIS.APPL.database.segment,@MIS.APPL.database.
directory),
C(:S,&S,?S),
ZZ%O(:,%["ilm.data"],&,$.data[:.CDFN],?,%["ilm.reqs.spec
s"]),
&LAB.TEM[TEST]^PNUM,
DO{+:LSPP[ACCT,DATE]^DATE ""^TIME,
DO{+:LSPP[ACCT,DATE,TIME]^TIME ""^PFX,
DO{+:LSPP[ACCT,DATE,TIME,PFX]^PFX ""^SPEC,
DO{+(:LSPP[ACCT,DATE,TIME,PFX,NUM],URN)^NUM
@CHECK.TESTS}}}},
""^ACCT^TEST^DATE^TIME^PFX^NUM^URN^A^B,
@MOST.RECENT,
C($U,%U,:U,&U,?U),
```

ABS to MRI using Z.link



Even speedier?

- Can we spare a prefix for the entire run of the report?
  - Most likely candidate, “&”
  - We can borrow it if we are not using any application dictionaries.

Application Specifics

- ABS , you can borrow * if not using any dx, proc, or drg names. you can borrow & if not using application dictionaries.
- ADM, you can borrow & or ? if not using MRI fields
- BAR, &, *
- Clinical Modules, less likely you can sneak away with a prefix..

ABS to MRI example

- Open and stack & in AL START macro with Z.link
- Use @Chg.prefix to force other database fields to translate with & prefix
- Get fields in detail macro
- Close and unstack & AL CLOSE.UP

ABS to MRI w/ Z.link

Step 1, open & to other database file

```
Enter/Edit Macro Logic: ABS.PAT.zcus.is

MRI."_(@.db#"1.")>DB,
DB>MIS.APPL.database,
C(&S),
%Z.link("&","MRI","%.MRI.data",DB)
```

Z.link / open 1x

- Step 2, get fields (use @Chg.prefix to force MRI.DRC fields to user &)

```
Enter/Edit Macro Logic: ABS.P

@Chg.prefix(MRI.DRC,?,&),
@mri.urn>MRI.PAT.medical.record,
@MRI.DRC.pat.street>STREET,
@MRI.DRC.pat.street2>STREET2,
@MRI.DRC.pat.city>CITY,
@MRI.DRC.pat.state>STATE,
@MRI.DRC.pat.zip>ZIP
```

AL D macro

@Chg.prefix

@Chg.prefix(DPM,oldpfx,newpfx)

```
Translating

Translating Logic: ABS.PAT.zcus.is.toronto.zlink.one.time.M.detail
IN:  @Chg.prefix(MRI.DRC,?,&),
OUT: "MRI.DRC ? prefix changed to &",
IN:  @mri.urn>MRI.PAT.medical.record,
OUT: :D2[dz]!5>ea,
IN:  @MRI.DRC.pat.street>STREET,
OUT: &AA[ea]AD!0>STREET,
IN:  @MRI.DRC.pat.street2>STREET2,
OUT: &AA[ea]AD!1>STREET2,
IN:  @MRI.DRC.pat.city>CITY,
OUT: &AA[ea]AD!2>CITY,
IN:  @MRI.DRC.pat.state>STATE,
OUT: &AA[ea]AD!3>STATE,
IN:  @MRI.DRC.pat.zip>ZIP
IN:
OUT: &AA[ea]AD!4>ZIP
Translation Time: 0
```

IMPORTANT!

- FIX (Restore) prefix at the end of report:

Crossing and Looping

@Next(subscript)

@Next(subscript,segment)

Translates to “plussing” (+) / logical nexting with subscript and assignment.

HUH?

@Next(dx.seq.no)

Translates to

+:DZ[dz]D["DX",dzDXN]^dzDXN

Demonstration of + operator

Structure:
/NAMES[“ABE”] = 1
/NAMES[“BOB”] = 1
/NAMES[“CARL”]=1
/NAMES[“DON”]=1
/NAMES[“EARL”]=1
And we write a line of code like this:
+/NAMES[SUB]^SUB
We get “ABE”

+ Operator

- + moves thru subscripts from lowest (nil) to highest, then back to nil.
- if you put value in a subscript, then + on it, you go to the next value
- after you get to the last subscript and + one more time, you get nil

DO syntax

DO{while true<space>do stuff}

List entire list of names:

DO{+/NAMES[SUB]^SUB N(SUB)^#}

DO syntax

- List partial list (A-D)

```
“ABE”^SUB,
DO{+/NAMES[SUB]^SUB<“EARL” N(SUB)^#}
```

Report Example

- ADM to PHA to get all active medications

Which field to link with?

@urn in ADM to @patient in PHA

Use index which starts with @patient

(same as you would do in frag from ADM)

Linking

- Linking issue – same as with fragments.
  - Look at indexes in the application you are going to, select one for which you have an efficient linking value in the application you are coming from.
- Extra step – you write the loops.

Data Definition

- **`@PHA.RX.patient.rx.x`** best index

```
pha.rx.patient.rx.index
  patient.rx.x

pha.rx.patient.type.index
  patient.type.x

pha.rx.patient.index
  patient.x
```

```
[patient,status,order.type,number] = urn
  :TRT[aa,trS,tdo,trN]

[patient,patient.type,urn]
  :TRV[aa,trP,tr]

[patient,number] = urn
  :TRP[aa,trN]
```

Report Picture

```
                Enter/Edit Report: Cross to PHA and get Active Meds          Page 3

Editing Picture - <ESC> for Fields or Regions     <Insert Mode>            [0->81]
REG  A  0---------1---------2---------3---------4---------5---------6---------7---------8-
HP      NAME                          MEDICATION                            DOSE
D    *  name_______________________   xx.med.name_________________________ xx.dose___
D       .                             xx.med.name_________________________ xx.dose___
        .
        +
```

MV Arrays

ADM to PHA – detail macro

```
change database to match your hospital
C(:S,?S,&S),
%Z.link(":","PHA","%.PHA.data","PHA.HHS.N"),
ZZ%O(?,:),
%Z.link("&","PHA","%.PHA.dic","PHA.HHS.N"),
@urn>PHA.RX.patient,
"AC">PHA.RX.status,
DO{@Next(PHA.RX.order.type,@PHA.RX.patient.rx.x) "">PHA.RX.number,
DO{@Next(PHA.RX.number,@PHA.RX.patient.rx.x) @LOAD}},
C(:U,?U,&S)

LOAD
@PHA.RX.patient.rx.x>PHA.RX.urn,
@PHA.DRUG.name[@PHA.RX.med]>/MV["D",CNT+1>CNT,2],
@PHA.RX.dose>/MV["D",CNT,3]

COMMENTS
;pha.rx.patient.rx.index          [patient,status,order.type,number] = urn
;patient.rx.x                     :TRT[aa,trS,tdo,trN]
```

Open data file

Open dictionary

restore status quo

Output

```
RUN DATE: 10/17/06                Hallmark **TEST**                          PAGE 1
RUN TIME: 1115              Cross to PHA and get Active Meds
RUN USER: IATRIC
NAME                        MEDICATION                          DOSE
TEST,RYAN                   TYLENOL                             650
                            BRETHINE,BRICANYL                   2.5
                            QUESTRAN POWDER                     0
```

Report example

- Go to LAB from ADM w/o fragment, get most recent GLU result for patient.

Field for linking..

- ADM to LAB
  - Use @urn from ADM to link to @patient in LAB (per visit approach)
  - Use @mri.urn from ADM to link to @mri.urn in LAB (all visits approach)

Data definition

lab.spec.l.mpi.index
[LAB.C.MRI.urn,collection.date,collection.time,prefix,number.part] = urn

mpi.x

lab.spec.l.patient.index
[patient,collection.date,collection.time,prefix,number.part] = urn

patient.x

Report Picture

```
Enter/Edit Report: Cross to Lab for Most Recent GLU                    Page 3

Editing Picture - <ESC> for Fields or Regions     <Insert Mode>          [0->81]
REG  A   0---------1---------2---------3---------4---------5---------6---------7---------8-
HP       NAME                          DATE      TIME RESULT     RANGE
D        name_________________________ xx.date_  xx.t xx.result_ xx.range_________________
TP
```

```
DAT=FREE
JFY=L
LEN=10
VAL=RESULT
```

Detail macro

```
Enter/Edit Macro Logic: ADM.PAT.zcus.is.toronto.cross.to.lab.M.detail  (AA: )

"LAB."_(@.db#"1.")>DB,
C(:S,?S),
%Z.link(":","LAB","%.LAB.data",DB),
ZZ%D(?,:),
@urn>LAB.L.SPEC.patient,
"">LAB.L.SPEC.collection.date,
DO{@Next(LAB.L.SPEC.collection.date,@LAB.L.SPEC.patient.x) "">LAB.L.SPEC.collection.time,
DO{@Next(LAB.L.SPEC.collection.time,@LAB.L.SPEC.patient.x) "">LAB.L.SPEC.prefix,
DO{@Next(LAB.L.SPEC.prefix,@LAB.L.SPEC.patient.x) "">LAB.L.SPEC.number.part,
DO{@Next(LAB.L.SPEC.number.part,@LAB.L.SPEC.patient.x) 1,
@LAB.L.SPEC.patient.x>LAB.L.SPEC.urn,
"100.0700">LAB.L.SPEC.test,
IF{@LAB.L.SPEC.status="COMP"&@LAB.L.SPEC.result 1,
@LAB.L.SPEC.collection.date>DATE,
@LAB.L.SPEC.collection.time>TIME,
@LAB.L.SPEC.result>RESULT,
@LAB.L.SPEC.normal.range>RANGE}}}}},
C(?U,:U)
```

Open LAB data file

Put fields in temp variables

Restore prefixes

Client Server Example

Data from:

ADM

NUR

PHA

LAB

OE

BAR

-- all on one report, no fragments

Bonus Info

Billing –

Just use acct.number from ADM and the account number index from BAR to find the BAR.PAT.account subscript.

Then the field BAR.PAT.balance will work.

(Along with all the other fields from BAR.PAT.main)

BAR

```
@acct.number^BAR.PAT.number,
""^BAR.PAT.account,
@Next(BAR.PAT.account,@BAR.PAT.number.x),
```

OE

Use patient (admissions urn) based index to get orders and put in MV array

Do two day date range.