MAGIC Syntax for NPR RW

Presented by:
Joe Cocuzzo
(978) 805-4115
Joe.Cocuzzo@iatric.com

About MAGIC

- Proprietary offshoot of MUMPS, aka “M”
- Developed by Dr. Octo Barnet with help from A. Neil Papalardo at Mass General
- Designed to be a lightweight language, particularly suited for string (text) data
- Popular for Medical applications such as Sunquest, Cerner and IDX

MEDITECH

Now in Four Flavors

- MAGIC - Proprietary OS/MAGIC language
- C/S - Windows OS/MAGIC language aka VMAGIC
- FS - Functional System. C/S platform, EMR & PCS
- FOCUS (M/AT 6.0) - EMR, PCS, POM, OE EDM move to FS with toolset. PCS data NOT returned to C/S

Coming Soon(ish)

New Focus Report Writer

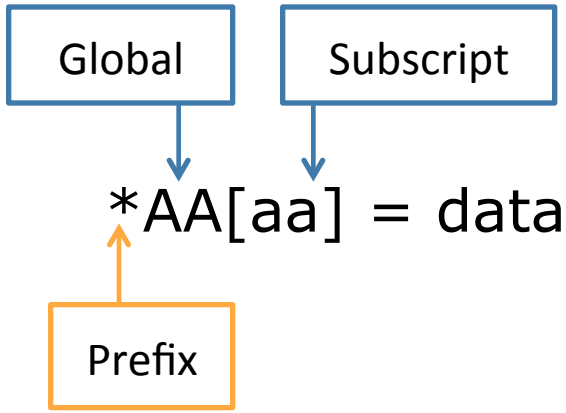
- Goals - No code required
- Reports from FOCUS and C/S databases
- C/S VMAGIC can get data from FOCUS with some complicated programming (iDAD)

Fundamentals of MAGIC

- LEFT to RIGHT evaluation
- Everything is a string
- Value = True
- No Value (nil) = False
- Powerful string operators (like MUMPS)
- Weak math skills
  10+2*5= 60 (!) 5/10 = 0 (!)

Fundamentals
Continued

Data automatically stored in tree



Data automatically sorted by subscripts

No need for searching or sorting algorithms

Fundamentals

Weaknesses

- Variable name + string < 256 in length
- Lines of code < 256 in length
- MAGIC only - 1K for variables and their values across all programs in session unless you stack symbol table

Symbol Table

In MAGIC

- 1024 bytes is all you get unless

  %ADM.PAT.zcus.is.your.macro.M.do(urn)X

- You get space back when you nil a variable

Symbol Table
Continued

- Symbol table limit applies to local variables only: STUFF, x, y, aa
- Not to slash variables

  /STUFF b.dis.date e.dis.date @.user
- b.dis.date translates to /b.discharge.date

  @.user translates to /.USR

Look at Data

```
$(A)["AA","E109","EV",20080131,1] = →ENADMIN→20080131→CT009→KAP.JER→KAP.JER→LYM1.IN→L104→1→L
                                    1.REHAB→L1.REHAB→PUL.REH→SP→→→AAR.BET
$(A)["AA","E109","EV",20080131,1,"D"] = →1336→AROTMAN→1336→L104-1
$(A)["AA","E109","EV",20080131,2] = →ENDISIN→20080131→→→→→→→→→→→Y→→→→→→→→E
$(A)["AA","E109","EV",20080131,2,"D"] = →1338→AROTMAN→1337
$(A)["AA","E109","G"] = →TEST,EXPIRED→SP→1 DALTON RD→→WALLINGFORD→CT→06492→(203)284-2800→045
                        -65-6565→→
$(A)["AA","E109","GE"] = →N/A→→→→→→→→RE
$(A)["AA","E109","I","SELFPAY"] = →1 DALTON ST→WALLINGFORD→CT→06492→(203)284-2800→SELF PAY→
$(A)["AA","E109","I","SELFPAY","E"] = →RE→N/A→
$(A)["AA","E109","I","SELFPAY","L"] = →VERIFIED→20080131→→AROTMAN
$(A)["AA","E109","I","SELFPAY","N"] = →TEST,EXPIRED→SP→→→→20080131→→→→→
$(A)["AA","E109","IO",1] = →SELFPAY
$(A)["AA","E109","NK"] = →TEST,HUSBAND→1 DALTON RD→→WALLINGFORD→CT→06492→(203)284-2800→→HU
$(A)["AA","E109","O3"] = →AROTMAN→→→→→→→→→→→→→→→→→→N
$(A)["AA","E109","O4"] = →→→→→→→→→→→→→→881069800
$(A)["AA","E109","PN"] = →TEST,HUSBAND→1 DALTON RD→→WALLINGFORD→CT→06492→(203)284-2800→→HU
```

```
*[AA,D1000000002] = →D000000001→ANDERSON,ANNE→SCH PPR→M006590→M457→19101010→19101010→F→97→→→→
HP8
*[AA,D1000000002,A] = →→→123-44-4555→→→→→→
*[AA,D1000000002,AD] = →100 MAIN STREET→→WESTWOOD→MA→02090→(123)123-1234→
*[AA,D1000000002,B] = →887243437→SCH MNR→→→→→→HP8→→→→→→→→→Y
*[AA,D1000000002,DR] = →→HAABR→→→→
*[AA,D1000000002,E] = →1ST BAPT→→420 S. ANDERSON→→ELWOOD→IN→46036→(765)552-2660
*[AA,D1000000002,EV,20080411,1] = →ENSCHPPR→20080411→→→→HP8→→
*[AA,D1000000002,EV,20080411,1,D] = →1038→MEDITECH→1040→→→→SCH PPR
*[AA,D1000000002,EV,20080412,1] = →EDSCHPPR→20080411→→→→HP8→→
*[AA,D1000000002,EV,20080412,1,D] = →0030→SCH MNR→1040→→→→SCH PPR
*[AA,D1000000002,EV,20080412,1,S,PPR] = 1
*[AA,D1000000002,O3] = →→20080411→→20080411→1038→→→→→→→→→→→N
*[AA,D1000000002,O4] = →→→→→→→→→→→→→→→→→→887193522
```

What Are These?

Data is packed or queued

```
*[AA,1] = →NEW
*[AA,D1000000002] = →D000000001→ANDERSON,ANNE→SCH PPR→M006590→M457→19101010→19101010→F→97→→
HP8
*[AA,D1000000002,A] = →→→123-44-4555→→→→→→
*[AA,D1000000002,AD] = →100 MAIN STREET→→WESTWOOD→MA→02090→(123)123-1234→
*[AA,D1000000002,B] = →887243437→SCH MNR→→→→→→HP8→→→→→→→→→Y
*[AA,D1000000002,DR] = →→HAABR→→→→
*[AA,D1000000002,E] = →1ST BAPT→→420 S. ANDERSON→→ELWOOD→IN→46036→(765)552-2660
*[AA,D1000000002,EV,20080411,1] = →ENSCHPPR→20080411→→→→HP8→→
*[AA,D1000000002,EV,20080411,1,D] = →1038→MEDITECH→1040→→→→SCH PPR
*[AA,D1000000002,EV,20080412,1] = →EDSCHPPR→20080411→→→→HP8→→
*[AA,D1000000002,EV,20080412,1,D] = →0030→SCH MNR→1040→→→→SCH PPR
*[AA,D1000000002,EV,20080412,1,S,PPR] = 1
*[AA,D1000000002,N3] = →→20080411→→20080411→1038→→→→→→→→→→→N
```

How to Make These Yourself?

Q(“FIRST”,”SECOND”,”THIRD”)^STUFF

- STUFF|0 = FIRST
- STUFF|1 = SECOND
- STUFF|2 = THIRD

Alternative Syntax

{“FIRST”,”SECOND”,”THIRD”}^STUFF

“THIRD”^STUFF|2

Add quotes automatically

`FIRST,SECOND,THIRD’^STUFF

Get Data Back Out

STUFF = FIRST^SECOND^THIRD

- STUFF|0=FIRST
- STUFF|1=SECOND
- STUFF|2=THIRD
- STUFF^{A,B,C}
- A=FIRST, B=SECOND, C=THIRD

So MAGIC - Easy Language

1) String operators
2) IF syntax
3) @Next, @Prev, + and –
4) DO syntax

MAGIC – Prefix management/Looping

C/S – Opening Database/Looping

String Operators

string at the position ABC#1 = B

$ to the left YYYYMMDD$4= YYYY

% to the right YYYYMMDD%5 = DD

` = not

ABC’#2 = AC

String Operators

YOURSTRING

- 0123456789
- YOURSTRING#3 = R
- YOURSTRING%3$3 = STR
- YOURSTRING’#3 = YOUSTRING

What is it Good For?

ER admissions by hour of the day:

- xx.hour
- DAT=INT
- LEN=2
- VAL=@service.time$2+0

Parsing Strings

Invented Delimiter

“BERMAN,JOEL F”#”0,” = BERMAN

“BERMAN,JOEL F”#”1,” = JOEL F

“BERMAN,JOEL F”#”1,”#”0<space>” = JOEL

String Operators

Parse Mnemonics

Mnemonics

- NUR.COCJ#”0,” = NUR
- IS.SMIF#”0.” = IS
- PURC.JOE#”0.” = PURCH

Strip SSN of Dashes

NNN-NN-NNNN’#3’#5

NNN-NN-NNNN’#3 = NNNNN-NNNN

01234567890

NNNNN-NNNN’#5

0123456789

(‘~ for general stripping)

MAGIC Math

Left to Right – no precedence of operation.
Decimal precision:

- + - * places = operand with most places
- / decimal places in numerator – places in denominator:

  5/10 = 0 (zero places – zero places = zero)

  5.0/10 = 0.5 (1 place – zero places = 1 place)

MAGIC Math

Rule of Thumb

X * 1.00000000000/Y :2D

Give X lots of places, then round with :nD
n = desired number of places with 5/4 rounding

More Punctuation ":"

: = format operator

:nD = round to n decimals

:nT = truncate to n characters

:nTL = truncate to n characters, left pad

:nTR = truncate to n characters, right pad

For zero padding: %Z.zero.fill() see your mouse pad for details

Lost Your Mouse Pad

Look at MT source code:

- Magic F(4) \Name of Program
- C/S F(5) DPM, then procedure
- Lookup available
- Arguments usually at top of program

Copy MEDITECH Macros

INIT report from standard

- "The more you need it, the less likely it is to work"
- List to report to paper
- Create report with same selects/ segments

Copy MEDITECH Macros

INIT report from standard

- Copy macros with F(4) MAGIC
- Exact name match needed in MAGIC F(5) C/S
- Put “M” at procedure prompt in C/S

Syntax Issues

- Take @ sign out of @Next subscripts
  DO{@Next(@dx) change to @Next(dx)
- Loop instead of Killing
  K(/STUFF) @Kill(/STUFF) $K(^/STUFF)
- DO{>/STUFF[SUB]^SUB “”^/STUFF
  [SUB]}

Syntax Issues

Continued

- Writing to @ 141^@Z.last.key
- Change to 141^/Z
- Writing to /. 1^@.pha.site ".PHA.SITE"^XXX, 1^/[XXX]

IF Syntax

IF{condition<space>statement}

IF{condition<space>statement;

Nextcond<space>statement;

Nextcond<space>statement;

Finalvalue}

IF Syntax

```
IF{@age.in.years^AGE<18 "Child";
AGE<65 "Adult";
AGE "Senior";
"No Age Available"}
```

Things That Mess Up IF

- Left hand value returned if comparison is true
- Nil returned if comparison is false
- Forgetting LEFT TO RIGHT (!)
- Not knowing that
  ! = minimum (not really OR)
  & = maximum (not really AND)

IF Failure

IF{@age.in.years^AGE<18!AGE>65 "Y”}

IF{17<18!17>65 "Y”}

IF{17!17 returns 17

IF{17>65 "Y"} fails

Solution: Add Parenthesis

IF{@age.years^AGE<18!(AGE>65) "Y”}

+ And -

- + is most essential operator in MAGIC
- Moves thru structures one subscript level at a time
- List next doctor (or first if DOC var is initially nil)

+\GU[DOC]^DOC

Sense of +

- + pushes subscript into variable
- Value of this expression goes from first doctor to last doctor, then to nil

+\GU[DOC]^DOC

Add a DO

Combine + and DO and you have a report writer

DO{+\GU[DOC]^DOC N(DOC)^#}

Syntax of DO

- DO{while.true<space>DOSTUFF}
- So list Doctors from start to "C"

DO{+\GU[DOC]^DOC<"D" N(DOC)^#}

List LI Vals

Report Example

Loop on c.location, build list of locations in string and print on report

@Next

Difference between + and @Next

- @Next is for structures in the Data Definition
- @Next(dx)
- @Next(room,@room.bed.index)

@Next

@Next uses data definition to figure out the subscripts and structure to loop on:

- @Next(dx)
- +?DZ[dz]DX[dxN]^dxN
- @Next(room,@room.bed.index)
- +:AARB[ggb,ggrB]^ggrB]
- @Next(subscript) or @Next (subscript,index)

Report Example

Scheduled and registered patients on same report

- Loop on Registration Index AND
- Loop on Scheduled Index "yourself"
- Put urns into list in slash

“Next Get” Syntax

+(/STUFF[SUB],DATA)^SUB

+ on subscript and get value of node in one operation

@Next.get(bed,@room.bed.index,urn)

Physical Next/Physical Previous

>:AARB[SUB]^SUB

>@room.bed.index[SUB]^SUB

SUB#0S = facility

SUB#1S = room

SUB#1S = bed

You can use > in Next Get syntax

>(@room.bed.index[SUB],DATA)^SUB

Physical Nexting

There is also physical previous (<)

No translator operator for > and <

i.e. nothing like: @Phys.Next(STUFF)

We Can Help!

Joe Cocuzzo

Vice President, NPR Services

Iatric Systems, Inc.

(978) 805-4115

Joe.Cocuzzo@iatric.com

Attend our free monthly webcasts.

Subscribe to our newsletter.

Iatric Systems, Inc.

Thank you.