Revise, Acquire, Automate! The Edit Language in NetView® for z/OS 6.1

Tom Howe, NetView Piper
IBM

Thursday, August 11, 2011
Session 9918
About Tom Howe

• In zNetView Design & Development for One Eon
  • If that means “27 years”
• Introduced NetView Pipelines, 1994
  • HT: John Hartmann
• Introduced WINDOW & EDIT, 1995
  • EDIT compares to CMS’s SPECS
• MRT, CRT, & Canzlog, lately
• With his wife – built house in Oxford, NC
  • Yes, with own hands
Edit a … what?

• From one message or command, take particular text, attributes, & literals…
  Build a new message or command.
  • No waiting, no change of environment, no scheduling
  • Fast!
• That’s the “why” in
  • automation’s ACQUIRE & EDIT,
  • PIPE EDIT,
  • MRT’s WHEN & REVISE,
  • CRT’s WHEN, REVISE, & WTO… more.
Two easy “edit scripts” – count ‘em

- WHEN (MSGID = ´IEE109I´)
  - The WHEN rule: two edits with = or ´¬= between
- MSGID, an input order, followed by default output order
  - NEXTWORD
- ´IEE109I´, a “delimited string”
  - Still followed by default output order

OK, just **HOW** is that helpful?

...Next page
New order: “last char of msgid”

- **NOT!** (this is not a Requirement, either)
- **WHEN** (MSGID **RIGHT 1** = ´E´)
  - On the left:
    - Input order **MSGID**
    - Conversion order **RIGHT 1**
    - Same default output as before

```
03 DSI802A CNM01
REPLY WITH ...  
```

```
DSI802A
```

```
RIGHT 1
```

```
A
```
The FORM of an Edit Script

- Global Order (optional)
  - E.g. PARSE, READLINE, RESETAUTO
- Edit phrase (optional)
  - Input order (get me something to work with)
    - E.g. word 4, AuthUser
  - Conversion order (change it)
    - E.g. C2X, OPDT, upCase
    - Repeat, as needed
  - Output order (put the text somewhere)
    - E.g. NEXT, WTOKEY, DISPLAY
- Repeat, as needed
An **AND** for the **WHEN** (MRT & CRT)

- When error msg AND “special user”
  rightmost msgid char is ‘E’
  AND
  AuthUser begins with ‘+’

- **WHEN** (MSGID RIGHT 1 1 1 AUTHUSER LEFT 1 N = ‘E+’)
  - Phrase 1 MSGID RIGHT 1 1
    - Might produce an ‘E’
  - Phrase 2 AUTHUSER LEFT 1 Next
    - Might produce a ‘+’ (as from +MASTER+ or +CONSOLE)
  - Two phrases, output abutted, could be ‘E+’

Not “awkward” - Power reuse; you get more function.
Multi-phrase edits can be *dense*

- Written straight out, multi-action edits can be hard to read

- So, *don’t do that*
  - All edit script implementations permit
    - multiple spaces
    - line breaks
    - Comments
      - *alt line for automation’s ACQUIRE & EDIT*
  - One “thought” per line? Phrase or Global item
    - CNME2011 has a EDIT spanning 81 source lines

A good read!
Edit Scripting I -- Pipelines

• EDIT stage reads & creates 1 (just 1) “message”
  • … at a time (Stream of messages in pipeline)
  • Message might be multi-line
  • If followed by NETV stage, output is a command

• Example scenario: a WTOR has been caught in a pipe
  • DIRVERT stage has prepended the next response line
    • See DIVERT help, COLLECT option
  • Result: a two line message consisting of
    text of response
    original WTOR text
Edit Scripting I – Pipelines cont.

- Starting with a two line message consisting of text of response original WTOR text
- EDIT "MVS REPLY" 1

  1. Literal string

  2. Where to put it

- Output line: MVS REPLY
Edit Scripting I – Pipelines cont.

- Starting with a two line message consisting of text of response original WTOR text

- EDIT "MVS REPLY" 1 LASTLINE

  Global: Move input source to bottom of MLWTO

- Output line (unchanged):
  MVS REPLY

continued
Edit Scripting I – Pipelines cont.

• Starting with a two line message consisting of text of response original WTOR text

• EDIT "MVS REPLY" 1 LASTLINE WORD 1 NW

From WTOR text, first “word” is reply ID.

• Output line:
  MVS REPLY 0106

NW short for NEXTWORD

continued
Edit Scripting I – Pipelines cont.

- Starting with a two line message consisting of text of response original WTOR text

- EDIT "MVS REPLY" 1
  LASTLINE
  WORD 1 NW
  "," NEXT

- Output line:
  MVS REPLY 0106,

Adding a short literal, as before
Edit Scripting I – Pipelines cont.

- Starting with a two line message consisting of original WTOR text

- EDIT "MVS REPLY" 1
  LASTLINE
  WORD 1 NW
  "," NEXT
  TOPLINE

- Output line (unchanged):
  MVS REPLY 0106,

  Global:
  Move input source to top of MLWTO
Edit Scripting I – Pipelines cont.

- Starting with a two line message consisting of text of response original WTOR text
- EDIT "MVS REPLY" 1 LASTLINE WORD 1 NW "," NEXT TOPLINE ALL NEXT
- Output line:
  MVS REPLY 0106,text of response

Next stage: NETV
Edit Scripting II – Automation Table

• ACQUIRE action uses edit script to produce text
  • *First line (only) of edit results*

• Example scenario, consider this message:

```
FTPDІnnn FTPXFER [stuff] RCODE=250
File=FTPXFЕR.PRD.МL.DI285.DI28510S
```

• Our task: Examine the highlighted fields; build command from values
• No REXX? Advantage: no allocate for REXXX environment

Adapted from online discussion at NetView Yahoo group
Edit Scripting II – AutoTbl partial MSGID

Ex: **FTPD1**nnn FTPXFER [stuff] RCODE=250
    File=FTPXFER.PRD.ML.DI285.DI28510S

- IF ACQUIRE(MSGID LEFT 5) = 'FTPD1' ...
**Edit Scripting II – AutoTbl Rtn Code**

**Ex:** `FTPD1nnn FTPXFER [stuff] RCODE=250
File=FTPXFER.PRDP.ML.DI285.DI28510S`

- IF ACQUIRE(MSGID LEFT 5) = 'FTPD1'
  & ACQUIRE('SKIPTO /RCODE=/ 7.4 strip') = '250'

Why 4? Need 250, not 2506.
Alt to PARSE & WORD

*continued*
Edit Scripting II – AutoTbl ‘PRD’ is ok

Ex: FTPD1nnn FTPXFER [stuff] RCODE=250
    File=FTPXFER.PRDM.L.DI285.DI28510S

- IF ACQUIRE(MSGID LEFT 5) = 'FTPD1'
  & ACQUIRE('SKIPTO /RCODE=/ 7.4 strip') = '250'
  & ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 3') ≠ 'TEST'

Counting from ‘File’, word 3 is second level qualifier, “PRD”
Edit Scripting II – AutoTbl JNAME for command

Ex: FTPD1nnn FTPXFER [stuff] RCODE=250
File=FTPXFER.PRD.ML.DI285.DI28510S

• IF ACQUIRE(MSGID LEFT 5) = 'FTPD1'
 & ACQUIRE('SKIPTO /RCODE=/ 7.4 strip') = '250'
 & ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 3') ≠ 'TEST'
 & ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 5') = JNAME

This SKIPTO is not a repeat!
Word 5, DI285, needed for Cmd

continued
Edit Scripting II – AutoTbl command

Ex: FTPD1nnn FTPXFER [stuff] RCODE=250
    File=FTPXFER.PR.D.ML.DI285.DI28510S

- IF ACQUIRE('FTPD1nnn') & ACQUIRE('RCODE=/ 7.4 strip') = '250'
  & ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 3') ≠ 'TEST'
  & ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 5') = JNAME
  THEN
    EXEC(CMD('MVS CHRIS 'JNAME' Etc'));

No, I don’t know the CHRIS cmd, either.
Edit Scripting II – Advantage, Summary

IF ACQUIRE(MSGID LEFT 5) = 'FTPD1'
& ACQUIRE('SKIPTO /RCODE=/ 7.4 strip') = '250'
& ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 3') ^= 'TEST'
& ACQUIRE('SKIPTO /File=/ PARSE /.=/ W 5') = JNAME THEN
  EXEC(CMD('MVS CHRIS 'JNAME' Etc'));

- Easier access to msg attributes, not normal “words”
- More natural, readable
- When cmds constructed, avoids REXX env overhead
- Allows EDIT of messages “in flight”
Edit Scripting III – Revision

• All revision orders used with
  • WHEN (both sides of the = or ñ=)
  • REVISE
  • WTO (cmd revision only)
• Same orders in both CRT and MRT
  • Except specifics like AUTOTOKEN & CONSAUTH
    • Special exception AuthUser & AuthGrp:
      • Not for CRT (apologies)
      • recedata(U) & recedata(G) in REXX

New for NV6.1

After NETVONLY action
What “unsupported” flag?

- All “other” flags and fields
- WQERQRC = Queue by Route Codes only
  - from WQE: offset D4 (212), mask=00000010 (listing!)
  
  
  
  WQE
  SUBSTR 213
  Byte at D4
  C2B
  11100010

- WHEN (WQE SUBSTR 213 C2B SUBSTR 7 = 1 )

Index = offset + 1
Edit Scripting III – Revision 128 routes & 32 descriptors

- FLGRTC[1-16] for input & output of route codes
- FLGDSC[1-4] for input & output of descriptor codes
  Both return 8 characters - 0s & 1s
- Example:
  REVISE('11xx0xxx' FLGRTCD1)
  • Set rt cd 1 and 2; clear rt cd 5; leave others alone
- Example:
  WHEN(FLGDSC2 SUBSTR 1 = 1)
  • Do this if descriptor code 9 is set
Edit Scripting III – Revision Using %INCLUDE within edit script

• MRT & CRT: any place you can put a blank –
  • You can put a line break
• At any line break – use %INCLUDE
• Example: WHEN ( %INCLUDE biglist )
• Biglist: 
  MSGID = ‘IEE123A’ | 
  MSGID = ‘IEF488E’ | …
  • Avoid duplication – easier maintenance

Not strictly about edit
Appplies in autotbl, too.
Edit scripting -- outlook

- Edit scripts used in four “big” environments
  - In multiple ways
- Flexible access to tokens, attributes
- Edit script compliers and interpreters reusable in NetView
- Any place text is handled, watch the two-fer principle
- Always remember: Requirements make us happy
  http://www.ibm.com/developerworks/rfe/
Questions?

??

The NetView group:
http://tech.groups.yahoo.com/group/NetView/