Intermediate Pipelines
VM 3D2

Will J. Roden
Endicott, NY
April 6, 2000

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "As is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk

.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead

.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Pipelines

Agenda — Pipelines

- Level Set
- A Journeyman's Tool Box
- Multistream Pipelines
- Selection Stages
- Its Looking UP
- Looking through our SPEC's
- Do it in REXX
- CallPipe
- REXX/Pipelines Interfaces

Level Set: Example — Pipelines

- Problem: For SEV=4 records, Sort by Dpt and display only Nbr, SEV, and Next Information
- Input File:

```
Nbr S Last      Ans          Next          Nx  Dpt
    E Act       Code         Action        Ac  Nxt
    V Done                   To Do         ID  Act
----+-------1----+--------2--------+-------3-----+--
001 3 RECEIVED               ANSWER        KL  A32
002 3 RECEIVED               ANSWER        BR  A32
003 3 RECEIVED               ANSWER        BI  A00
004 5 ANSWERED   COLD         INT1          VM  B29
005 3 ANSWERED   DUP          CLOSE         CO  B26
006 3 OPENED                 RECEIVE       BI  A00
007 4 REROUTED               RECEIVE       PE  B18
008 3 OPENED                 RECEIVE       BI  A00
009 4 RECEIVED               ANSWER        BE  B18
010 3 ANSWERED   COLD         SSV           DE  A31
011 4 CLOSED     CNEW         NONE
012 4 CLOSED     CNEW         NONE
013 3 OPENED                 RECEIVE       CP  A64
014 4 RECEIVED               ANSWER        NE  B18
015 1 OPENED                 RECEIVE       CP  A64
016 3 RECEIVED               ANSWER        XX  A31
```

Level Set: Example.... Pipelines

Level Set: Example..... Pipelines

- **Pipe Command:**

```
Pipe   < INPUT FILE A| drop 4| locate 5.1 /4/
|   sort 34-36 | specs 1.3 1 5.1 nw 22.16 nw
|   > OUTPUT FILE A
```

- **Pipe Command in REXX EXEC:**

```
/* TEST1 EXEC                                          */
'Pipe ( endchar ?)',
   '   < INPUT FILE A',           /*Read   */
   '|  drop 4',                  /* Select */
   '|  locate 5.1 /4/',          /*Select */
   '|  sort 34-36',              /*Sort   */
   '|  specs 1.3 1 5.1 NextWord 22.16 nw',
   '|  > OUTPUT FILE A'        /*Write    */
Exit
```

- **Output:**

011 4 NONE
012 4 NONE
007 4 RECEIVE PE B18
009 4 ANSWER BE B18

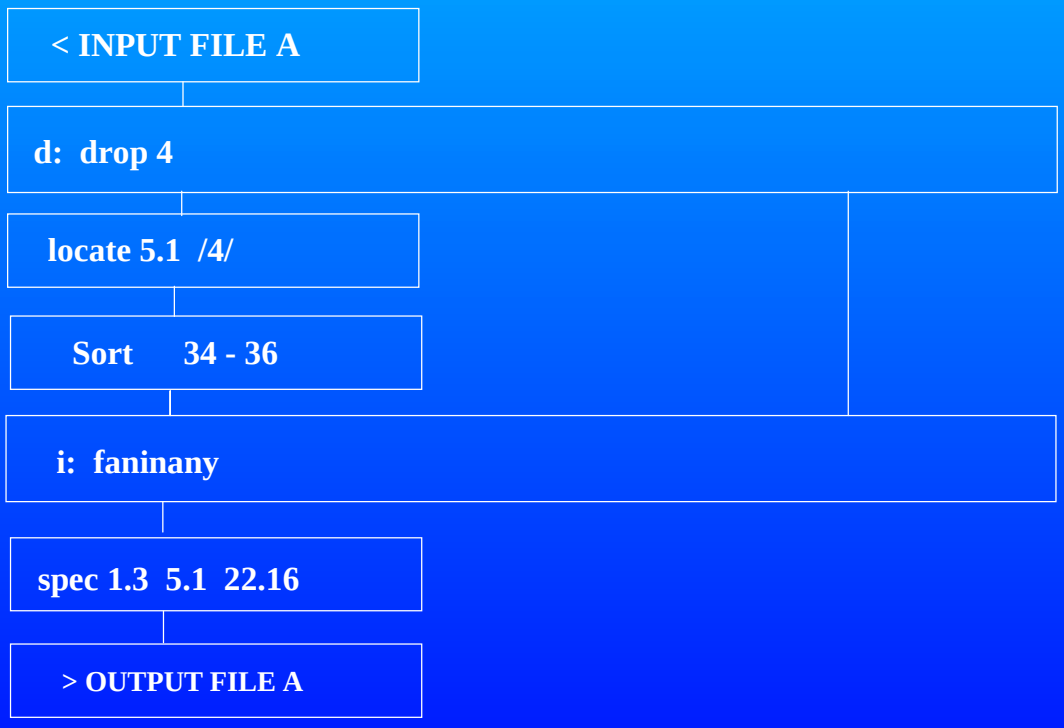
The Tool Box — Pipelines

- How to get HELP
  - HELP - From CMS Pipeline Reference
  - AHELP - Author's help
    - To use:

      pipe (a) help menu
      pipe (a) help locate
      help pipe menu
      help pipe locate
      help pipe

  - CMS Pipelines User's Guide - SC24-5777
  - CMS Pipelines Reference - SC24-5778
  - CMS Pipelines Author's Edition- SL26-0018
  - Quick Reference - SX24-5290

Multistream Pipelines — Pipelines



Multisteam Pipelines .... Pipelines

- RPTHDG EXEC

```
/* RPTHDG EXEC    4/24/92                    */
 'Pipe ( endchar ?)',
   '    < INPUT FILE A',       /* Read       */
   '|d:drop 4',                /* Select     */
   '|   locate 5.1 /4/',
   '|   sort 34-36',           /* Sort,Pick */
   '|i:faninany',
   '|   spec 1.3 1 5.1 NextWord 22.16 nw',
   '|   > OUTPUT FILE A ',     /* Write      */
   '?d:',                     /* Keep Hdg   */
   '|i:'
  Exit
```

- Output data for RPTHDG EXEC -OUTPUT FILE A

```
Nbr S Next    Nx Dpt
  E Action   Ac Nxt
  V To Do    ID Act
--- + ---+----3----+-
011 4 NONE
012 4 NONE
007 4 RECEIVE  PE B18
009 4 ANSWER   BE B18
014 4 ANSWER   NE B18
```

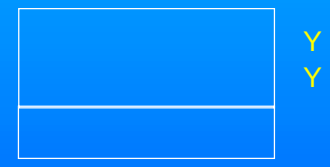
Selection Stages Pipelines

- Flow Diagram

- Divides one stream into two streams
- Does NOT delay the record
- Stage ends when all output streams are disconnected

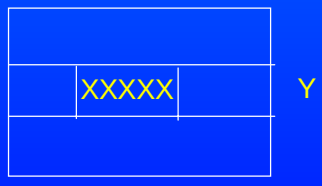
Selection Stages

Pipelines

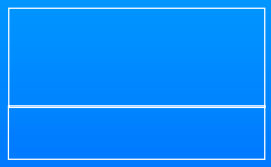


TAKE 2

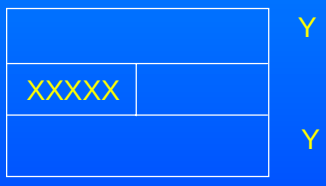
FIND
STRFIND



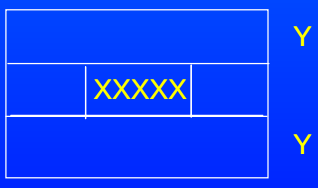
LOCATE, ALL, or PICK



DROP 2



NFIND
STRNFIND



NLOCATE

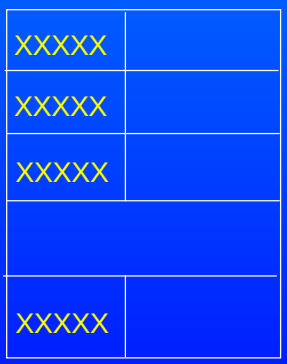
Y - record selected

Selection Stages ...

Pipelines

TO LABEL
STRTOLABEL
TOTARGET

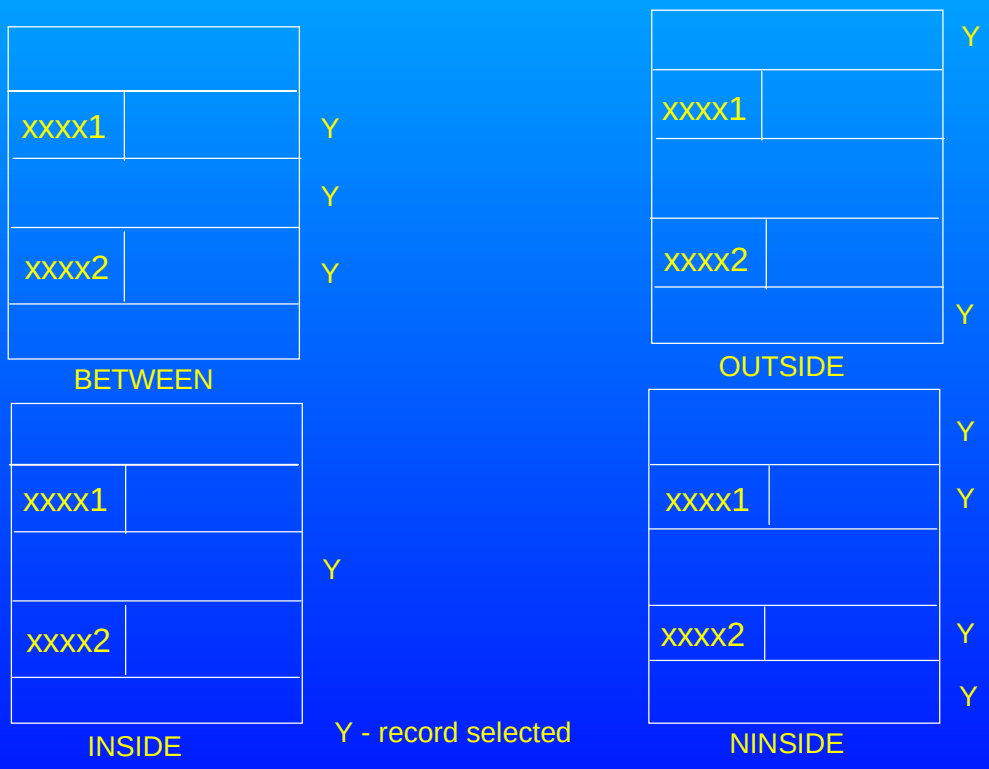
FRLABEL
STRFRLABEL
FRTARGET
FROM TARGET



WHILELABEL and STRWHILELABEL

Y - record selected

Selection Stages ... Pipelines



Selection Stages Pipelines

- Selection Stage prefixes
  - NOT
    - Reverses output streams of a stage

```
Pipe   < MATH FILE
   |   not all (/1/ ! /56/) & x4b
   |   console
234 - 567
345 * 675.
```

    - CASEI
      - Allows a selection stage to function without regard to case (CASE Insensitive)

```
pipe    literal a b c A B C
   |     split
   |     casei locate /a/
   |     console
a
A
```

Selection Stages ... Pipelines

- Selection Stage prefixes
  - ZONE
    - Allows a selection to use a range of columns

```
pipe    literal  a b c  A B C
        | zone 3.1 between /b/  2
        | console
a b c  A B C
```

  - REVERSE
    - Reverses the contents of the record

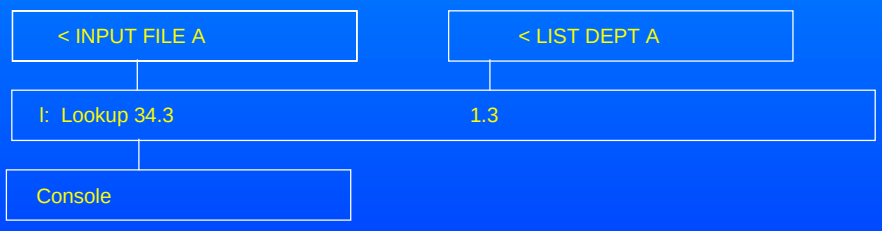
```
pipe    literal 123 abc
        | reverse
        | strfind  /cba/       /* find  'abc'    */
        | reverse
        | console
123 abc
```

Using Lookup Pipelines

- Get Depts to match from a file
- Input data for LOOKUP EXEC -LIST DEPT A

```
A32
A31
```

- Lookup with Detail and Master



- Output

```
001 3 RECEIVED            ANSWER    KL    A32
A32 Process
002 3 RECEIVED            ANSWER    BR    A32
A32 Process
010 3 ANSWERED COLD       SSV       DE    A31
A31 Design
016 3 RECEIVED            ANSWER    XX    A31
A31 Design
```

Using Lookup .... Pipelines

- Same input file - INPUT FILE A

```
Nbr S Last      Ans    Next     Nx Dpt
    E Act       Code   Action   Ac Nxt
    V Done             To Do    ID Act
----+----1----+----2----+----3----+--
001 3 RECEIVED        ANSWER    KL A32
002 3 RECEIVED        ANSWER    BR A32
003 3 RECEIVED        ANSWER    BI A00
004 5 ANSWERED COLD   INT1      VM B29
005 3 ANSWERED DUP    CLOSE     CO B26
006 3 OPENED          RECEIVE   BI A00
007 4 REROUTED        RECEIVE   PE B18
008 3 OPENED          RECEIVE   BI A00
009 4 RECEIVED        ANSWER    BE B18
010 3 ANSWERED COLD   SSV       DE A31
011 4 CLOSED   CNEW   NONE
012 4 CLOSED   CNEW   NONE
013 3 OPENED          RECEIVE   CP A64
014 4 RECEIVED        ANSWER    NE B18
015 1 OPENED          RECEIVE   CP 464
016 3 RECEIVED        ANSWER    XX A31
```

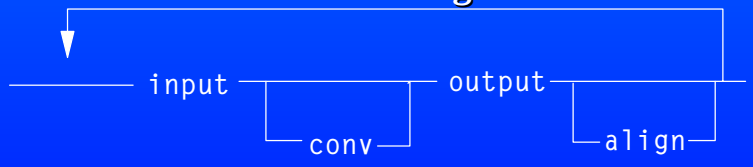
Using Lookup Pipelines

- LookUp EXEC

```
/* Lookup Example                                        */
'Pipe ( end ?)',
  '    < INPUT FILE',                        /* Read List */
  '|l:lookup 34.3 1.3',
  '|   console',                             /* Write     */
  '?   < LIST DEPT',                  /* Get Dept List    */
  '|l:'                          /* 2nd I/P of Lookup     */
Exit
```

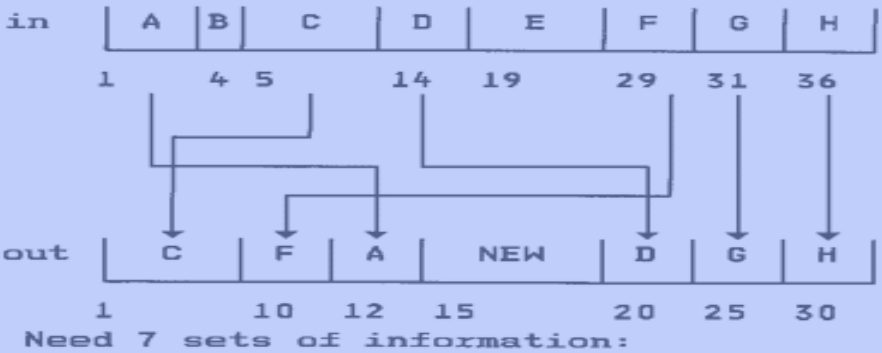
SPECs Pipelines

- SPECS constructs output records field by field
- Output fields can come from one or more input records, literal values, record number or time of day clock
- SPECS
  - Rearranges record contents
  - Converts data into different formats
  - Aligns or centers data in output field
  - Strips leading or trailing blanks
  - Pads output fields
- Basic Format of SPECS stage command



SPECs ... Pipelines

◆ For Example ...



Need 7 sets of information:

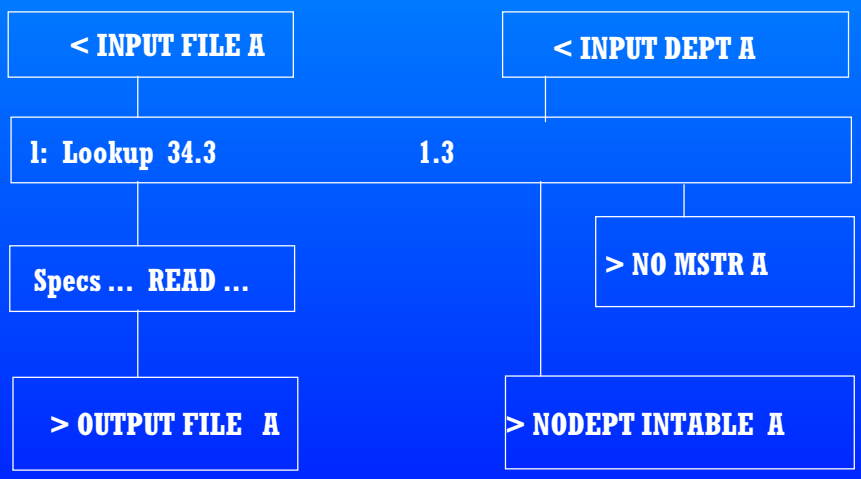
	FROM	TO	What?
SPECS	1–3	12	A
	5.9	1	C
	14–18	20	D
	/NEW/	15	
	29.2	10	F
	31.5	25	G
	36–*	30	H

SPECs Pipelines

- Additional Features of SPECs
  - ►READ
    - – Read the next record now so its data can be used
    - – Used to combine records
    - – Ex: Specs 1.3 1 5.1 NW 22.16 nw read 5-* nw
  - ►WRITE
    - – Write the built record now so another one can be built
    - – Used to create additional records
    - – Ex: Specs 1.3 1 5.1 NW write 1.3 1 22.16 nw
  - ►SELECT n
    - – Read data from STREAM n
    - – Used to combine data from multiple streams
    - – Ex: Specs 1.3 1 5.1 NW select 1 1.3 nw
  - ►WORDSEParator (WS)
    - – Specify what separates words
    - – NULL words are not counted
    - – Ex: Specs ws - w1 1 W5 nw
  - ►FIELDSEParator (FS)
    - – Specify what separates fields
    - – No default
    - – NULL fields are counted

SPECs Pipelines

- Lookup with Detail and Master



- LookUp EXEC

```
/* Lookup Example with SPECs READ                    */
'Pipe ( end ?)'.
   '    < INPUT FILE'.          /* Read              */
   '|1:lookup 34.3 1.3'.        /* Dept name         */
   '|   > lookup temp a'.       /* For Demo          */
   '|   spec 1.3 1 5.1 NextWord 22.16 nw'.
   '        read 5-* nw'. /* Combine Rods           */
   '|   > OUTPUT FILE A '.      /* Write             */
   '?   < INPUT DEPT'.
   '|1:'.              /* 2nd I/O of Lookup         */
   '|   > NODEPT INTABLE A'.
   '?1:'.             /* 3rd O/P of Lookup         */
   '|   > NO MSTR A'
Exit
```


- Output data for LOOKUP EXEC - OUTPUT FILE A

```
001  3  ANSWER    KL  A32  Process
002  3  ANSWER    BR  A32  Process
003  3  ANSWER    BI  A00  Code
004  5  INT1      VM  B29  Test
005  3  CLOSE     CO  B26  Finance
006  3  RECEIVE   BI  A00  Code
007  4  RECEIVE   PE  B18  Code
008  3  RECEIVE   BI  A00  Code
009  4  ANSWER    BE  B18  Code
010  3  SSV       DE  A31  Design
014  4  ANSWER    NE  B18  Code
016  3  ANSWER    XX  A31  Design
```

- Output data for LOOKUP EXEC - NO_DEPT IN_TABLE A

```
011  4  CLOSED    CNEW   NONE
012  4  CLOSED    CNEW   NONE
013  3  OPENED           RECEIVE   CP  A64
015  1  OPENED           RECEIVE   CP  A64
```

- Output data for LOOKUP EXEC - NO MSTR A

```
XXX Not Found
```

Writing Stages in REXX Pipelines

- Reading Pipeline Records
  - ►Peekto
  - ►Readto
- Writing Pipeline Records
  - ►Output
- Select - Pick and Input/Output Stream
- Maxstream - Highest stream that is hooked up
- Streamnum - What stream is selected?
- Streamstate - What is happening to my selected stream?
- Stagenum - Where is this stage in the Pipeline?
- Short - Hook my selected I/P to my selected O/P
- Message - Tell them this NEW message
- IssueMsg - Tell them this EXISTING message

Improved MsgEvery REXX Stage Pipelines

- Do MsgEvery with conditional secondary output stream
  - Write all input records the primary output stream
  - Write a message with the record count every n records
  - If the secondary output stream is connected, write the message to it, otherwise to a SAY
  - Make sure this stage is NOT first in the Pipeline

```
/* Do a MSG for every COUNT lines */
 parse arg count message       /* get args  */
'maxstream output'   /* 2 output steams?    */
 Stream Count = rc

'stagenum'  /* What Stage are we?  */
 StagePosition = rc
 if (StagePosition = 1) then do
  'issuemsg 127  FPLMYM'
   say '_____________________________________'
  'message Do Not put me first'
   say '_____________________________________'
  'message FPLMYM127E Don not put me first'
   signal BigError
end
```

Improved MsgEvery REXX Stage Pipelines

```
 signal on error
'PEEKTO in_data '
 do rcd_nbr = 1 while rc = o
    'OUTPUT' in_data
     if (rcd_nbr // count = 0) then do
        if (StreamCount < 2) then do
           say message "record Number:' rcd_nbr
        end
        else do
           'select 1'
           'output' message 'Record Number: 'rcd_nbr
           'select 0'
        end
     end
   'READTO'
   'PEEKTO in_data '
end /* do while rc = 0 */

error:
   say message ' Record Number:' rcd-nbr
BigError:
   Exit RC * (RC <> 12)
```

Improved MsgEvery REXX Stage ... Pipelines

```
/* MSGEVERY — NO Secondary Stream  */
say '====== No Secondary Stream ============='
'Pipe'.
   '    < INPUT FILE'.
   '|   msgevery 3 This is an Example'.
   '|   > OUTPUT FILE A'
say '====== Has Secondary Stream ============'
/* MSGEVERY with Secondary Stream  */
'Pipe (endchar ?)'.
   '    < INPUT FILE'.
   '|m:msgevery 3 This is an Example'.
   '|   > OUTPUT FILE A'.
   '?m:'.
   '|   console'
say '====== First In Pipeline — Message ===='
/* MSGEVERY — as First Stage  */
'Pipe'.
   '    msgevery 3 This is an Example'.
   '|   > OUTPUT FILE A'
```

Improved MsgEvery REXX Stage..... Pipelines

- **MSGEVERY Console Log**

```
=====No Secondary Stream ==============
THIS IS AN EXAMPLE   Record Number: 3
THIS IS AN EXAMPLE   Record Number: 6
THIS IS AN EXAMPLE   Record Number: 9
THIS IS AN EXAMPLE   Record Number: 12
THIS IS AN EXAMPLE   Record Number: 15
THIS IS AN EXAMPLE   Record Number: 18
THIS IS AN EXAMPLE   Record Number: 20
=====Has Secondary Stream ================
THIS IS AN EXAMPLE   Record Number: 3
THIS IS AN EXAMPLE   Record Number: 6
THIS IS AN EXAMPLE   Record Number: 9
THIS IS AN EXAMPLE   Record Number: 12
THIS IS AN EXAMPLE   Record Number: 15
THIS IS AN EXAMPLE   Record Number: 18
THIS IS AN EXAMPLE   Record Number: 20
```

Improved MsgEvery REXX Stage ... Pipelines

```
====== First In Pipeline — Message ====
DMSMYM2771E Stage command cannot be
      specified as the
DMSMYM2771E first stage of a pipeline
DMSPMG2653I ... Issued by stage number 1
      of pipeline number 1
DMSPMG2651I ... Running stage: msgevery 3
      This is an Example
_____________________________________
Do not put me first.
_____________________________________
FPLMYM127E Do not put me first.
DMSPMG2653I ... Issued by stage number 1
      of pipeline number 1
DMSPMG2651I ... Running stage: msgevery 3
      This is an Example
```

Example of Pipe in a Pipe Pipelines

Example of Pipe in a Pipe Pipelines

- Sortex REXX

```
/* To sort the records of a pipeline, but leaving a
specified number of records in the front of the file alone (not sorted).
SAMPLE:  SORTEX 3  1.6  11-30                    */

arg qty sort_args
'Callpipe ( endchar  ? )',
   '    *:',
   '|t:take  'qty,
   '|i:fanin',
   '|  *:',
   '?t:',
   '|  sort 'sort_args,
   '|i:'
Exit

/* To demonstrate SORTEX     */
'Pipe ( endchar ?)',
   '  < INPUT FILE',
   '| SORTEX 4 34.3 ',
   '| > OUTPUT FILE A'
Exit
```

Example of Pipe in a Pipe

Pipelines

OUTPUT FILE A:

```
Nbr S Last      Ans          Next           Nx  Dpt
    E Act       Code         Action         Ac  Nxt
    V Done                   To Do          ID  Act
----+-------1----+--------2--------+-------3-----+--
011 4 CLOSED    CNEW         NONE
012 4 CLOSED    CNEW         NONE
003 3 RECEIVED               ANSWER         BI  A00
006 3 OPENED                 RECEIVE        BI  A00
008 3 OPENED                 RECEIVE        BI  A00
010 3 ANSWERED   COLD        SSV            DE  A31
016 3 RECEIVED               ANSWER         XX  A31
001 3 RECEIVED               ANSWER         KL  A32
002 3 RECEIVED               ANSWER         BR  A32
013 3 OPENED                 RECEIVE        CP  A64
015 1 OPENED                 RECEIVE        CP  A64
007 4 REROUTED               RECEIVE        PE  B18
009 4 RECEIVED               ANSWER         BE  B18
014 4 RECEIVED               ANSWER         NE  B18
005 3 ANSWERED  DUP          CLOSE          C0  B26
004 5 ANSWERED  COLD         INTA           VM  B29
```

Obtaining REXX Variables

Pipelines

- **VAR**
  - ► **Gets/Sets a Variable**
- **STEM**
  - ► **Gets/Sets a stemmed array**
- **REXXVARS**
  - ► **Gets all variables**
- **VARLOAD**
  - ► **Sets all variables**
- **PRODUCER option**
  - ► **Obtains or sets REXX variables from the stage preceding it in the pipeline**
  - ► **Consider: pipe ONE | TWO | console**
    - – **TWO can obtain or set variables in ONE**
  - ► **The following must exist:**
    - – **Stage ONE must be blocked in output**
    - – **Stage TWO must issue the command from a CALLPIPE**

Acronyms and IBM Trade Marks — Pipelines

- **Acronyms**
  - CMS - Conversational Monitor System
  - CP - Control Program
  - IUCV - Inter-user Communication Vehicle
  - TCP - Transmission Control Protocol
- **IBM Trade Marks**
  - VM/ESA - Virtual Machine/Enterprise Systems Architecture

For More Information — Pipelines

- CMS Pipelines User's Guide - SC24-5777
- CMS Pipelines Reference - SC24-5778
- CMS Pipelines Author's Edition - SL26-0018
- Quick Reference - SX24-5290
- Development Contacts:
  - Will J. Roden, Jr.
    - Phone: (607) 752-5065
    - Internet: RODEN@US.IBM.COM
    - Web: http://www.vm.ibm.com/devpages/roden
  - US Mail
    - IBM Department G37G
    - 1701 North Street
    - Endicott, NY 13760