





Stages Built Into
NetRexx Pipelines 5.01
&
CMS Pipelines V7R1
and Their Differences

How to Read Syntax Diagrams

Special diagrams (often called railroad tracks) are used to show the syntax of external interfaces.

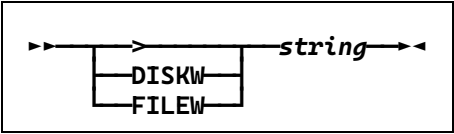
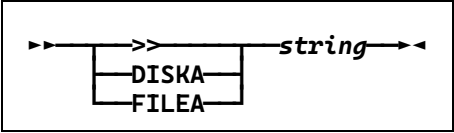
To read a syntax diagram, follow the path of the line. Read from left to right and top to bottom.

- The  symbol indicates the beginning of the syntax diagram.
- The  symbol, at the end of a line, indicates that the syntax diagram is continued on the next line.
- The  symbol, at the beginning of a line, indicates that the syntax diagram is continued from the previous line.
- The  symbol indicates the end of the syntax diagram.

Within the syntax diagram, items on the line are required, items below the line are optional, and items above the line are defaults.

Pipelines Builtin Stages

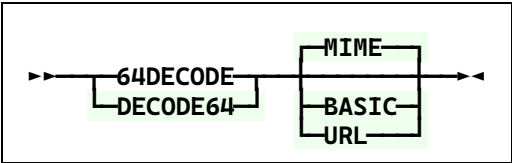
Show Stages Implemented in: <input checked="" type="checkbox"/> NetRexx Pipelines: <input checked="" type="checkbox"/> CMS Pipelines:
Show All Details: <input checked="" type="checkbox"/> (Double click on a row to turn it on/off.)
<input checked="" type="checkbox"/> Highlight NetRexx Only / CMS Only:
Show Only Changes: <input checked="" type="radio"/> None <input type="radio"/> 5.01 <input type="radio"/> 4.06 <input type="radio"/> 4.05 <input type="radio"/> 3.11 <input type="radio"/> 3.09

> <i>diskw</i> <i>filew</i> 3.09	Replace or Create a File  <ul style="list-style-type: none">• delegates to diskw.
>> <i>diska</i> <i>filea</i> 3.09	Append to or Create a File  <ul style="list-style-type: none">• delegates to diska.
>> <i>mdsk</i>	Append to or Create a CMS File on a Mode <ul style="list-style-type: none">• Not implemented in NetrexX Pipelines.
>> <i>mvs</i>	Append to a Physical Sequential Data Set <ul style="list-style-type: none">• Not implemented in NetrexX Pipelines.
>> <i>oe</i>	Append to or Create an OpenExtensions Text File <ul style="list-style-type: none">• Not implemented in NetrexX Pipelines.
>> <i>sfs</i>	Append to or Create an SFS File <ul style="list-style-type: none">• Not implemented in NetrexX Pipelines.
>> <i>sfsslow</i>	Append to or Create an SFS File <ul style="list-style-type: none">• Not implemented in NetrexX Pipelines.

>m_{dsk}	Replace or Create a CMS File on a Mode <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
>m_{vs}	Rewrite a Physical Sequential Data Set or a Member of a Partitioned Data Set <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
>o_e	Replace or Create an OpenExtensions Text File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
>s_{fs}	Replace or Create an SFS File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
< diskr filer 3.09	Read a File <div data-bbox="279 584 734 714" data-label="Diagram"> <pre> graph LR In[] --> Box subgraph Box [] DISKR FILER end Box --> Out[string] style In fill:none,stroke:none style Out fill:none,stroke:none </pre> </div> <ul style="list-style-type: none"> Implemented as in CMS; delegates to diskr.
<m_{dsk}	Read a CMS File from a Mode <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<m_{ys}	Read a Physical Sequential Data Set or a Member of a Partitioned Data Set <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<o_e	Read an OpenExtensions Text File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<s_{fs}	Read an SFS File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<s_{fsslow}	Read an SFS File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
-- comment 3.09 5.01	Comment Stage, No Operation <div data-bbox="279 1467 866 1574" data-label="Diagram"> <pre> graph LR In[] --> COMMENT COMMENT --> Out[string] style In fill:none,stroke:none style Out fill:none,stroke:none </pre> </div> <ul style="list-style-type: none"> delegates to comment. Not in CMS Pipelines; This is a STAGE, not a programming comment. It must have a SPACE after ---. It must have either a stageEnd or pipeEnd. If ended with a stageEnd, it passes records through on primary input to output streams. If ended with a pipeEnd, it does NOT pass records through. If used before a driver stage, it must have a pipeEnd.
3277b_{fra}	Convert a 3270 Buffer Address Between Representations <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
3277e_{nc}	Write the 3277 6-bit Encoding Vector <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.

64decode
decode64
3.11

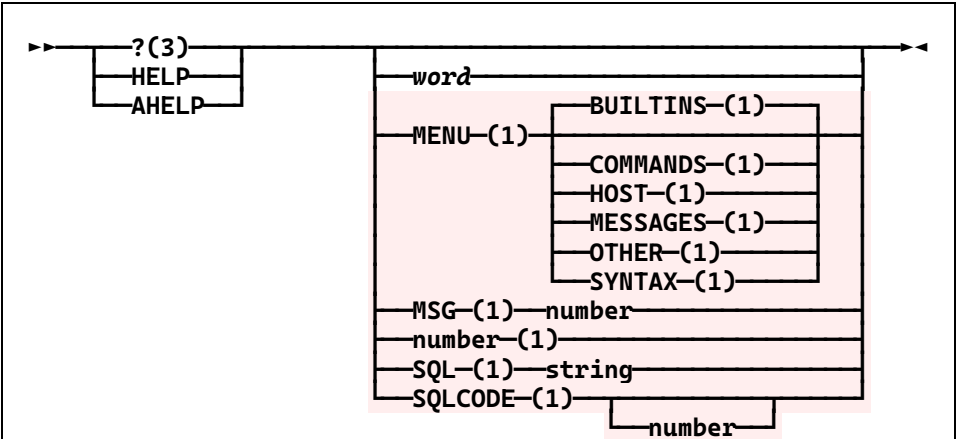
Decode Base-64 Format



- NOTE: CMS is only 64DECODE, and does not have the options; it does MIME.
- BASIC - Output is mapped to a set of characters lying in A-Za-z0-9+/. The encoder does not add any line feed in output, and the decoder rejects any character other than A-Za-z0-9+/.
- URL - Output is mapped to set of characters lying in A-Za-z0-9+_. Output is URL and filename safe.
- MIME - Output is mapped to MIME friendly format. Output is represented in lines of no more than 76 characters each, and uses a carriage return '\r' followed by a linefeed '\n' as the line separator. No line separator is present to the end of the encoded output.
- 3.11: New to NetRexx. Add MIME, BASIC, & URL options.

?
ahelp
help

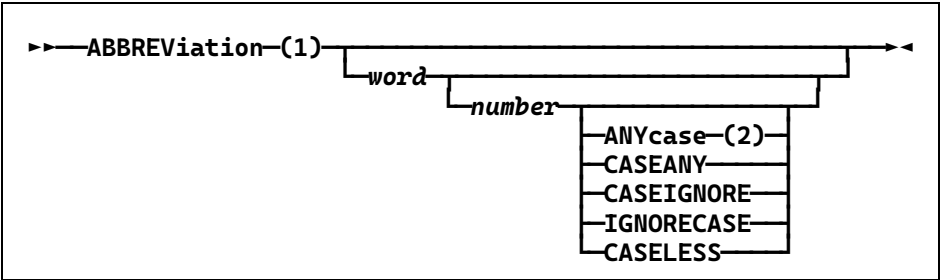
Display Help for Pipelines



- (1) CMS Pipelines only. Not yet in NetRexx Pipelines.
(2) If primary output is connected, lines are propagated, otherwise they are sent to the console by "say."
(3) ? is the default pipeEnd character. Here it is useful only when a different pipeEnd is defined.

abbreviation
abbreviatio
abbreviati
abbreviat
abbrevi
abbrev

Select Records that Contain an Abbreviation of a Word in the First Positions



- (1) ABBREVIation must be ABBREV in CMS
- (2) ANYcase must be ANYCASE in CMS

acigroup

Write ACI Group for Users

- Not implemented in Netrex pipelines.

addrdw

Prefix Record Descriptor Word to Records

- Not implemented in Netrex pipelines.

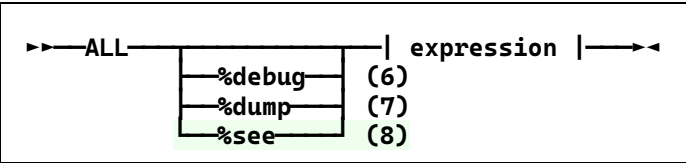
adrspace

Manage Address Spaces

- Not implemented in Netrex pipelines.

<i>aftfst</i>	Write Information about Open Files <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>aggrc</i>	Compute Aggregate Return Code <div> ▶▶—AGGRC—◀◀ </div>
<i>ahelp</i> <i>help</i> <i>?</i>	Display Help for Pipelines <div> <p>(1) CMS Pipelines only. Not yet in NetRexx Pipelines.</p> <p>(2) If primary output is connected, lines are propagated, otherwise they are sent to the console by "say."</p> <p>(3) ? is the default pipeEnd character. Here it is useful only when a different pipeEnd is defined.</p> </div>

Select Lines Containing Strings (or Not)



- Notes:
- (1) "expression" consists of one or more delimited strings separated by logical ANDs, ORs, and NOTs, and grouped, if needed, by parentheses.
 - (2) "&" is used for AND.
 - (3) Since "|" is the default stage separator, "!" may be used for OR.
 - (4) Since NetRexx uses "(" and ")" for options -- which are not used in the ALL stage -- "[" and "]" must be used for parentheses.
 - (5) CMS Pipelines, having originated on 3270 terminals, uses "~" for NOT. This symbol is not readily typed on terminals running NetRexx Pipelines, so as alternatives, "\", used by NetRexx, (it needs to be doubled to "escape" it) or "^", used by KEX, NOT symbols may be used as alternatives.
 - (6) %debug (must be lowercase) NetRexx Pipelines writes the logic line to the file ALL.DEBUG in the current directory. Windows may make it all.debug . CMS Pipelines writes the constructed pipeline (of LOCATE and NLOCATE stages) to ALL DEBUG A.
 - (7) %dump (must be lowercase) - writes to the primary output stream as the first record. NetRexx Pipelines writes the logic line. CMS Pipelines writes constructed pipeline.
 - (8) %see (must be lowercase) - NetRexx Pipelines Only. Writes the logic line to the standard output (terminal).
 - CMS Pipelines uses is own logic order. NetRexx Pipelines uses regular NetRexx logic.

Examples:

```

• literal NetRexx is Good,NetRexx is Great,NetRexx is Fantastic |
  split , |
  all /a/ |
  cons

  ►NetRexx is Great
  ►NetRexx is Fantastic

• literal NetRexx is Good,NetRexx is Great,NetRexx is Fantastic |
  split , |
  all / G/ & [/oo/ ! /F/] |
  cons

  ►NetRexx is Good

• literal NetRexx is Good,NetRexx is Great,NetRexx is Fantastic |
  split , |
  all /R/ & [/oo/ ! /F/] |
  cons

  ►NetRexx is Good
  ►NetRexx is Fantastic

```

<i>alter</i>	<p>Change the contents of records, from one character to another</p> <p>NetRexx</p> <ul style="list-style-type: none"> • An variation on the theme of Xedit's ALTER. • There are pairs of char1s and char2s, optionally separated by commas. • For each pair of char1 and char2, this changes ALL char1s to char2, like Xedit's 4th and 5th parameters were 1 and 1. • The chars can be single characters, 2 or 3 digit decimal numerical representations, or beginning with H, h, X, or x hexadecimal representations. • Also see TRANSLATE / XLATE. • Not in CMS Pipelines.
<i>alserv</i>	<p>Manage the Virtual Machine's Access List</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>apldecode</i>	<p>Process Graphic Escape Sequences, Old APL language</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>aplenode</i>	<p>Generate Graphic Escape Sequences, Old APL language</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>append</i>	<p>Put Output from a Device Driver after Data on the Primary Input Stream</p>

►►—ARRAY—*netrexx_array_name*—◄◄

- Pipes for NetRexx
- If this is the first stage in a pipe, it reads the *netrexx_array_name* from the parent NetRexx program and puts the members into the pipe one record at a time.
- If this is any stage but the first in a pipe, it writes the records to the *netrexx_array_name* in the parent NetRexx program.
 - This ERASES the *netrexx_array_name* before writing the first record.
 - Except for this, this stage is identical to ARRAYA.

Examples:

```
class testpipe
method testpipe()
  F = Rexx 'abase'      -- Two NetRexx stem variables of objects
  T = Rexx 1            -- with default type Rexx, we needn't specify it

  F[0]=5                -- Fill the first stem variable
  F[1]=222
  F[2]=3333
  F[3]=1111
  F[4]=55
  F[5]=444

  pipe (apipe stall 1000 )
    stem F |            -- read a NetRexx array to objects (records) |
    sort |              -- sort column-wise |
    console |           -- show them sorted |
    stem T ?            -- write them to the NetRexx array

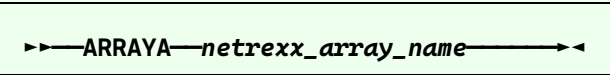
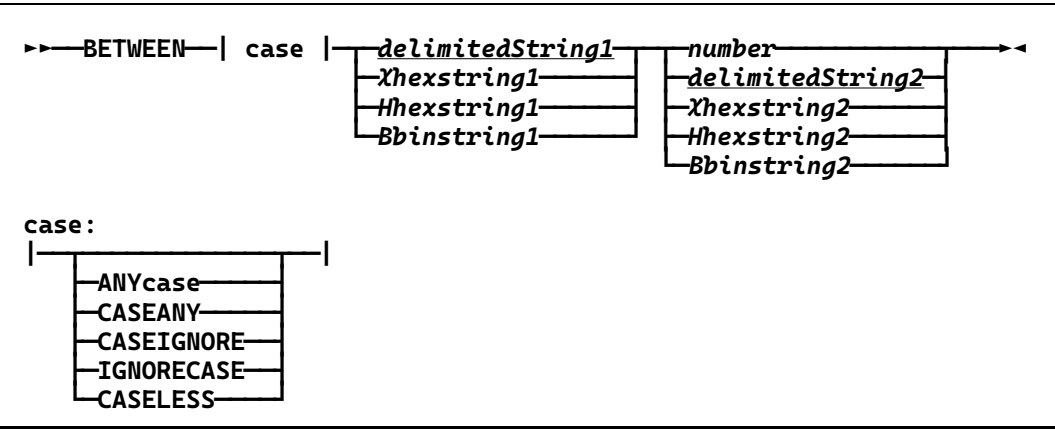
  loop i=1 to T[0]
    say 'T['i']='T[i]
  end

  exit

method main(a=String[]) static
  testpipe()

====
PS > pipc testpipe.nrx
pipe (testpipe_apipe stall 1000 ) stem F | sort | console | stem T
PS > java testpipe

1111
222
3333
444
55
T[1]=
T[2]=1111
T[3]=222
T[4]=3333
T[5]=444
T[6]=55
PS >
```

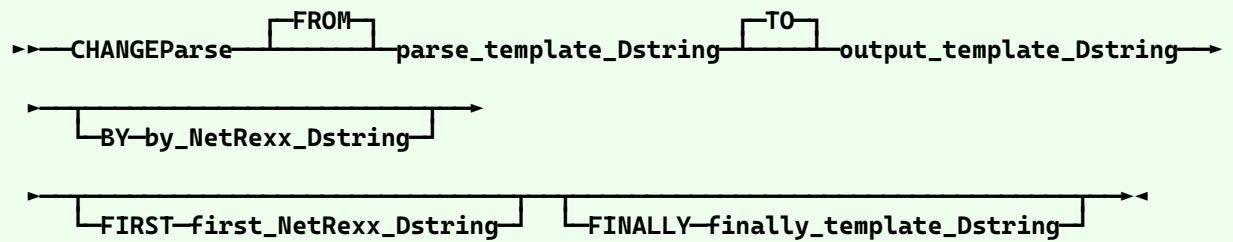
<i>arraya</i>	Read or Addto Write an Array <div>  </div> <ul style="list-style-type: none"> • Pipes for NetRexx • If this is the first stage in a pipe, it reads the <i>netrex_array_name</i> from the parent NetRexx program and puts the members into the pipe one record at a time. • If this is any stage but the first in a pipe, it adds the records to the <i>netrex_array_name</i> in the parent NetRexx program. <ul style="list-style-type: none"> ◦ This ADDS to the <i>netrex_array_name</i>. ◦ Except for this, this stage is identical to ARRAY. <p>Examples: See ARRAY.</p>
<i>asatomc</i>	Convert ASA Carriage Control to CCW Operation Codes. Old printer control <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>asmcont</i>	Join Multiline Assembler Statements <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>asmfind</i>	Select Statements from an Assembler File as XEDIT Find <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>asmnfind</i>	Select Statements from an Assembler File as XEDIT NFind <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>asmxpnd</i>	Expand Joined Assembler Statements <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>beat</i>	Mark when Records Do not Arrive within Interval <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>between</i> 3.09	Select Records Between Labels <div>  </div>
<i>bfs</i> <i>hfs</i>	Read or Append File in the Hierarchical File System <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>bfsdirectory</i> <i>bfsdir</i> <i>hfsdirectory</i> <i>hfsdir</i>	Read Contents of a Directory in a Hierarchical File System <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>bfsquery</i> <i>bfsq</i> <i>hfsquery</i> <i>hfsq</i>	Write Information Obtained from OpenExtensions into the Pipeline <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>bfsreplace</i> <i>bfsrep</i> <i>hfsreplace</i> <i>hfsrep</i>	Replace the Contents of a File in the Hierarchical File System <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.

<i>bfsstate</i> <i>bfsstat</i> <i>hfsstate</i> <i>hfsstat</i>	Obtain Information about Files in the Hierarchical File System <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>bfsxecute</i> <i>bfsx</i> <i>hfsxecute</i> <i>hfsx</i>	Issue OpenExtensions Requests <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>block</i>	Block to an External Format <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>browse</i> <i>brw</i>	Display Data on a 3270 Terminal <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>buffer</i>	Buffer Records <div> <pre> graph LR BUFFER --> number number --> delimitedString </pre> </div>
<i>c14to38</i>	Combine Overstruck Characters to Single Code Point. Old printer.
<i>casei</i>	Run Selection Stage in Case Insensitive Manner <div> <pre> graph LR CASEI --> ZONE ZONE --> inputRange inputRange --> one1["(1)"] one1 --> REVERSE REVERSE --> one2["(1)"] stage --> string </pre> </div> <ul style="list-style-type: none"> (1) CMS Pipelines only.
<i>change</i>	Substitute Contents of Records <div> <pre> graph LR CHANGE --> anycase anycase --> inputRange inputRange --> range range --> FROMTO["FROMTO-(1)"] FROMTO --> changeString changeString --> FROM["FROM-(1)"] FROM --> delimitedString1["delimitedString"] delimitedString1 --> TO["TO-(1)"] TO --> delimitedString2["delimitedString"] delimitedString2 --> FOR["FOR-(1)"] FOR --> numorstar </pre> <p>changeString: delimiter string delimiter string delimiter </p> <p>anycase: RESPECTCASE-(1) ANYcase CASEANY CASEIGNORE IGNORECASE CASSLESS </p> </div> <ul style="list-style-type: none"> (1) NetRexx Pipelines only.

changeparse
 changepars
 changepar
 changepa
 changep
 3.11

Change the contents of records, using Rexx Parse. Calculations can be done.

NetRexx



- Records are parsed via the parse_template_delimited_string.
- Variables are named \$n, where n is 1 to 9.
- The by_NetRexx_delimited_string is interpreted. This is 0 or more semicolon separated NetRexx statements, probably using the \$n variables, which can have the value altered.
- Other variables may be used, and are persistent while the stage is active, so can be used as accumulators.
- The values of the variables are put into the output_template_delimited_string replacing \$n.
- For a literal \$n that won't be changed, use \$\$n.
- A first_NetRexx_delimited_string, if present, is interpreted before reading any record from the primary input stream. This is 0 or more semicolon separated NetRexx statements, probably using the \$n variables. Any variables used in the by_NetRexx_delimited_string must be defined here.
- A finally_template_delimited_string, if present, is written as a final output record after the primary input stream is finished, using the \$n's.
- Any keyword phrases must, in any order, follow any non-keyworded FROM & TO phrases.
- This is NetRexx Pipelines only, not in CMS.

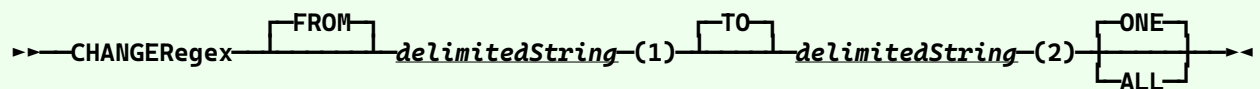
Examples:

- changeparse / 2 \$1 +1/ /The second letter is "\$1". \$\$1 won't be changed./
- changeparse from / 2 \$1 +1/ to /The second letter is "\$1". \$\$1 won't be changed./
- changeparse from / . \$2 . 50 \$5 +5 / to /The product is \$1/ by /\$1 = \$2 * \$5/
- changeparse from / . \$2 . 50 \$5 +5 / ,
 to /The product is \$1/ ,
 by /\$1 = \$2 * \$5;\$3 = \$3 + \$1/ ,
 first /\$3 = 0/ ,
 finally /\$3 is the total/

changeregex
 changerege
 changereg
 changere
 changer
 3.09

Substitute Contents of Records using Java Regular Expressions

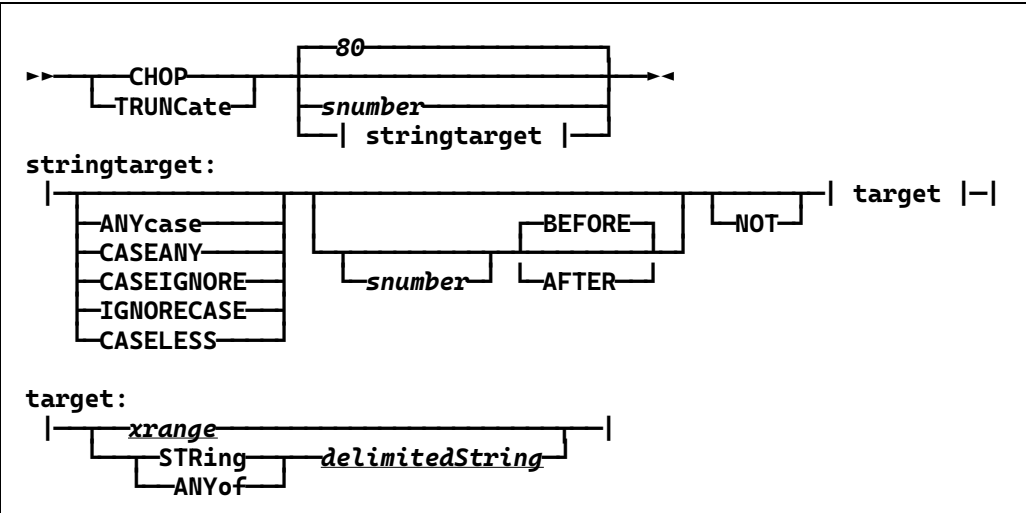
NetRexx



- Uses the Java RegEx classes and its dialect of RegEx. See Java's **Pattern** class and **replaceFirst** and **replaceAll** methods of **String** for full documentation.
- (1) First, FROM, delimitedString is a Java *RegEx expression* for what is to be replaced.
- (2) Second, TO, delimitedString is the replacement string. It may contain elements from the first one.
- This is NetRexx Pipelines only, not in CMS.

chop
truncate
truncat
trunca
trunc

Truncate the Record



cipher

Encrypt and Decrypt Using a Block Cipher

- Not implemented in Netrexx Pipelines.

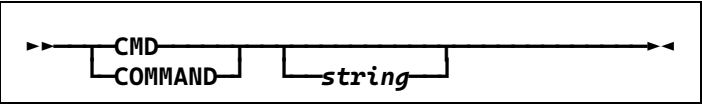
ckddeblock

Deblock Track Data Record

- Not implemented in Netrexx Pipelines.

cmd
command

Issue OS Commands, Write Response to Pipeline

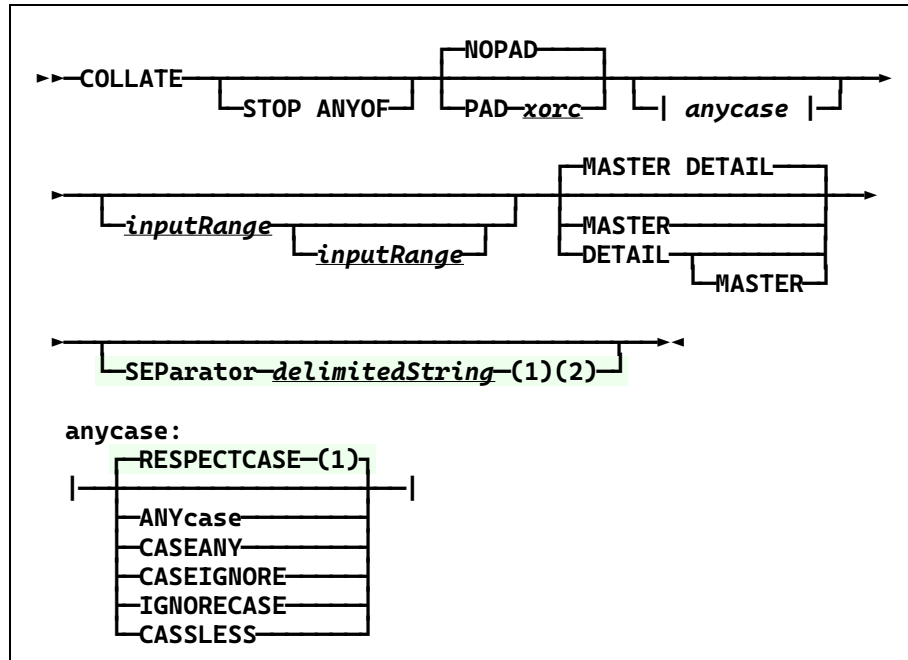


- input stream 0 is for commands
- input stream 1 is stdin
- output stream 0 is stdout
- output stream 1 is the return code
- output stream 2 is stderr

cms

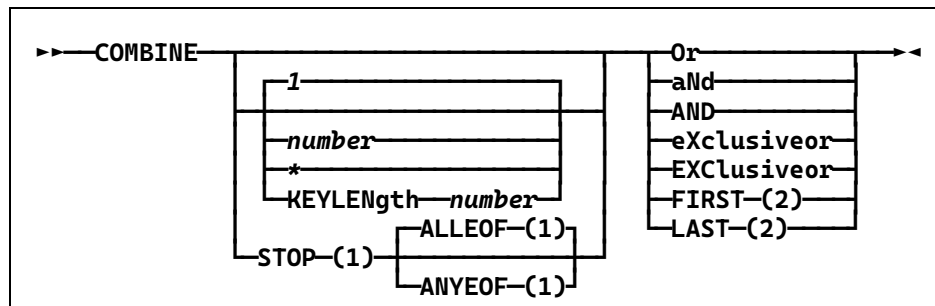
Issue CMS Commands, Write Response to Pipeline

Collate Streams



- (1) NetRexx Pipelines only.
- (2) delimitedString record is put before each Master Record (or after if DETAIL MASTER order) on the primary output stream.
- 3.11 New to NetRexx Pipelines. Add SEPARATOR option.

Combine Data from a Run of Records

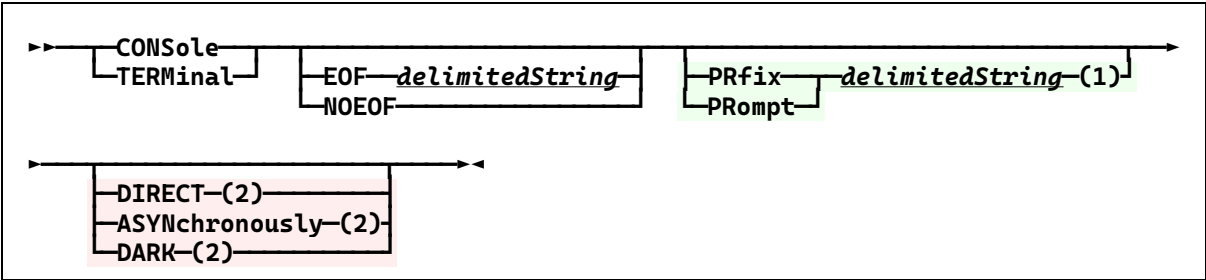


- (1) Only for use with secondary input streams. Only options from this column usable with any secondary input streams. (This is poorly documented in CMS Pipelines. This is a best guess of their intentions.)
- (2) Not usable with STOP and secondary streams.

command cmd	Issue OS Commands, Write Response to Pipeline <div data-bbox="280 129 986 232"> </div> <ul style="list-style-type: none"> • input stream 0 is for commands • input stream 1 is stdin • output stream 0 is stdout • output stream 1 is the return code • output stream 2 is stderr
comment -- 3.09 5.01	Comment stage <div data-bbox="280 479 896 591"> </div> <ul style="list-style-type: none"> • Not in CMS Pipelines; • This is a STAGE, not a programming comment. It must have a SPACE after --. • It must have either a stageEnd or pipeEnd. • If ended with a stageEnd, it passes records through on primary input to output streams. • If ended with a pipeEnd, it does NOT pass records through. • If used before a driver stage, it must have a pipeEnd.
compare	Compare Primary and Secondary Streams, Write the Result <div data-bbox="280 869 1445 1218"> </div> <ul style="list-style-type: none"> • (1) -1 = Primary is shorter/less, 0 = equal, 1 = Secondary is shorter/less • (2) 0 = equal, 1 = not equal • (3) Primary is LESS/shorter (or MORE/longer) than secondary • (4) <i>DStrings</i> can use any of the following escapes (or the lowercase) for the unequal situation: <ul style="list-style-type: none"> ◦ \C (count) for the record number, ◦ \B (byte) for column number ◦ \P (primary) for the primary stream record ◦ \S (secondary) for the secondary stream record ◦ \L (Least) for the stream number that is shorter, -1 if equal ◦ \M (Most) for the stream number that is longer, -1 if equal • (5) Equal or not, this <i>DString</i> precedes any of the others. • (6) This is NetRexx Pipelines only, not included in CMS • (7) In reporting \P & \S, control characters, except new line, \n, are transliterated to [blob, 219.d2c()] • (8) Without ECHO, this stops and reports at first non-compare. With ECHO, each primary input is reported; after first non-compare primary input stream records continue to be read and reported, but no testing is done. • (9) Options work in any order • Input streams: <ul style="list-style-type: none"> ◦ 0: Data 1 ◦ 1: Data 2 • Output streams: <ul style="list-style-type: none"> ◦ 0: Result (single record, possibly multiple lines) ◦ 1: Last primary record read at first no match, or end of stream ◦ 2: Last secondary record read at first no match, or end of stream
configure	Set and Query CMS Pipelines Configuration Variables <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.

console
consol
conso
cons
terminal
termina
termin
termi
term
3.11

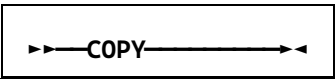
Read or Write the Terminal in Line Mode



- (1) NetRexx only
On first stage, delimitedString is put out as a prompt
On other stages, each line is prefixed with delimitedString
Outout to next stage does NOT include delimitedString
Either keyword can be used for either stage
- (2) CMS only

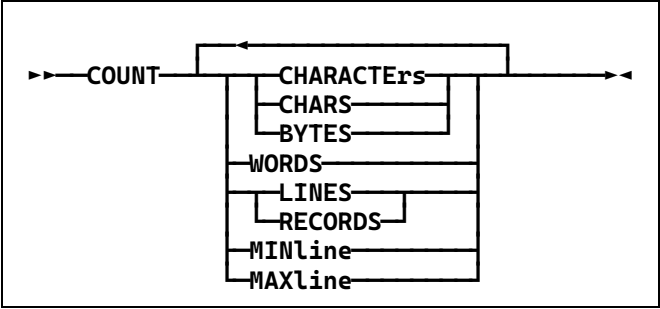
copy

Copy Records, Allowing for a One Record Delay



count

Count Lines, Blank-delimited Words, and Bytes



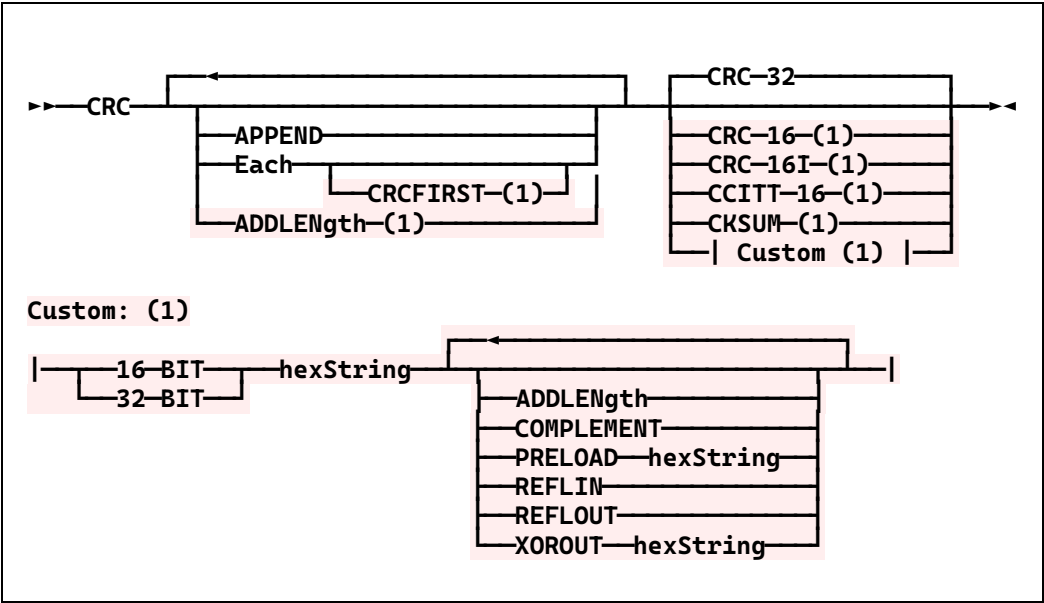
cp

Issue CP Commands, Write Response to Pipeline

- Not implemented in Netrex Pipelines.

crc
4.05

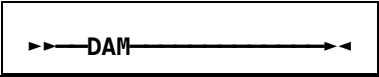
Compute Cyclic Redundancy Code



- (1) Not implemented in Netrexx Pipelines.
- (2) CRC stage uses secondary output, if connected.

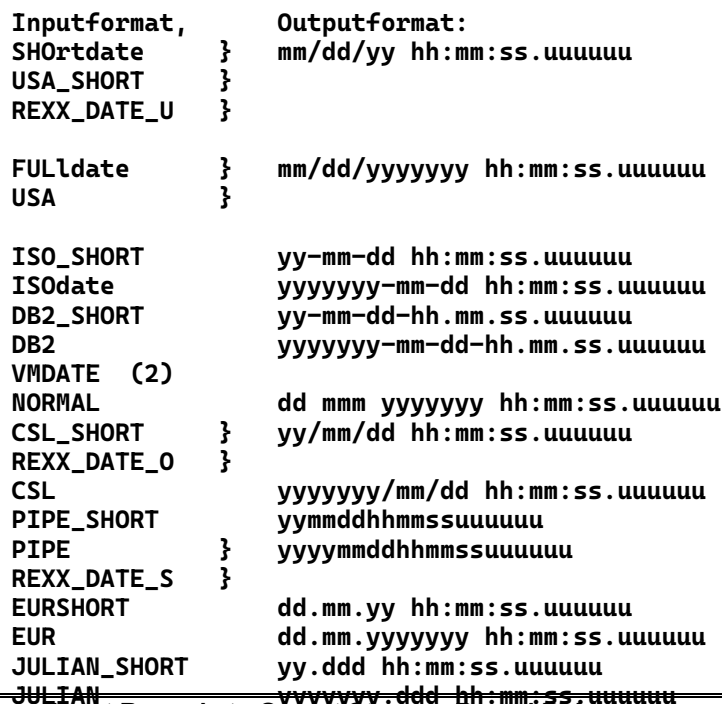
dam

Pass Records Once Primed

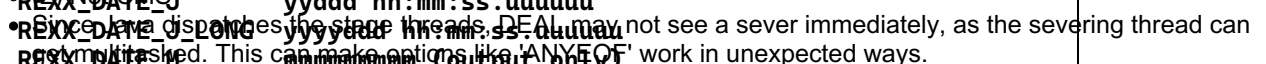


dateconvert
dateconver
dateconve
dateconv
3.09

Convert Date Formats

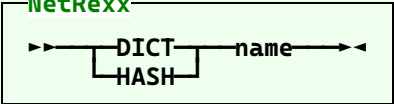
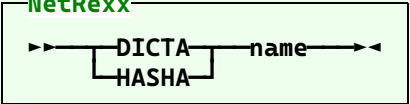
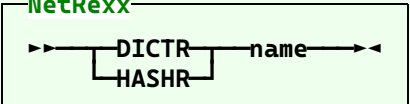
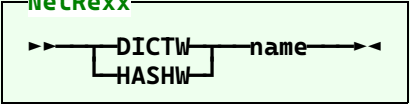
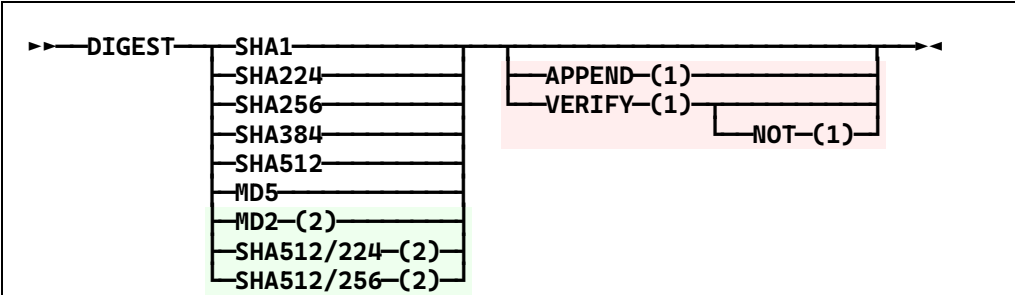
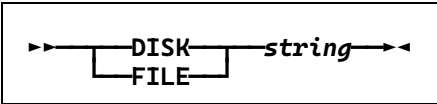


PETRO ABSOLUTE } (3)tput Streams Round Robin



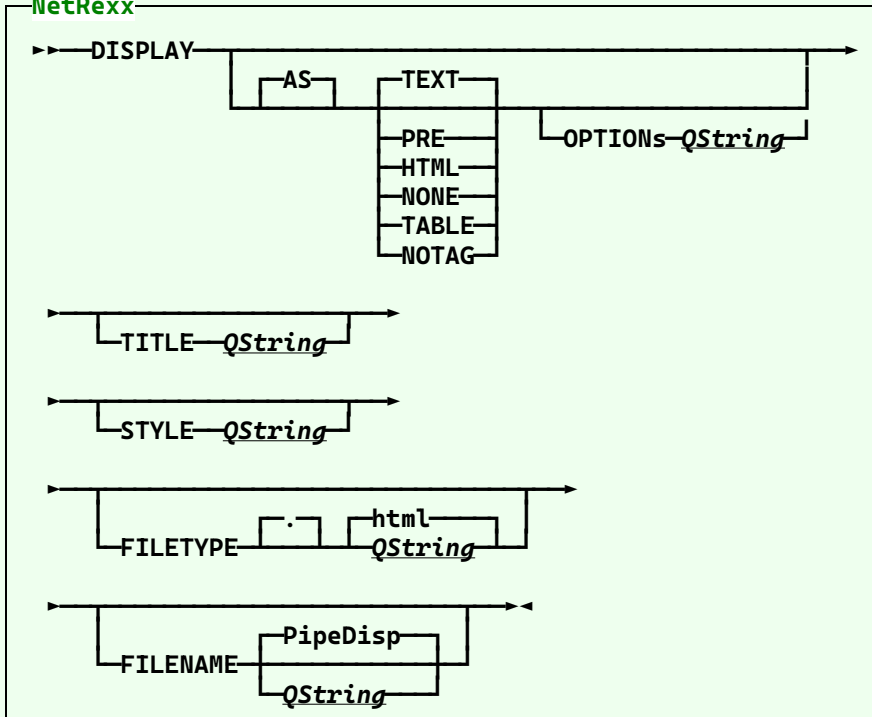
- (1): SPACE is optional here.
- (2) Not implemented in NetRexx Pipelines at this time; mainly mainframe useful only.
- (3): NetRexx Pipelines uses IRange which gives a superset of range options.
- (4): NetRexx Pipelines only. What time to assume if blank time on input.
- (5): NetRexx Pipelines only.
 - Use current local date time.

<i>deblock</i>	Deblock External Data Formats <div data-bbox="280 129 1327 436"> </div> <ul style="list-style-type: none"> • CMS has many more mainframe centric formats that NetRexx Pipelines does not process. • (1) Not CMS Pipelines
<i>decode64</i> <i>64decode</i> <i>3.11</i>	Decode Base-64 Format <div data-bbox="280 604 794 766"> </div> <ul style="list-style-type: none"> • NOTE: CMS is only 64DECODE, and does not have the options; it does MIME. • BASIC - Output is mapped to a set of characters lying in A-Za-z0-9+/. The encoder does not add any line feed in output, and the decoder rejects any character other than A-Za-z0-9+/. • URL - Output is mapped to set of characters lying in A-Za-z0-9+_. Output is URL and filename safe. • MIME - Output is mapped to MIME friendly format. Output is represented in lines of no more than 76 characters each, and uses a carriage return 'r' followed by a linefeed 'n' as the line separator. No line separator is present to the end of the encoded output. • 3.11: New to NetRexx. Add MIME, BASIC, & URL options.
<i>delay</i> <i>4.05</i>	Suspend Stream <div data-bbox="280 1108 1513 1296"> </div> <ul style="list-style-type: none"> • (1) Arguments are NetRexx Pipelines only, not CMS. CMS (and NetRexx when there is no argument) reads delays as the first word of each record. When arguments are present, they follow the CMS conventions for the delay time in records. The + indicates a duration, no + means time of day. The objects do NOT have the delay as the first word. Clock hours are 24h, so 2pm is 14, and are for the next 24 hours if before "now." Seconds can have decimal point and milliseconds. In relative times, the number of seconds and minutes are not limited to 60; so 120 seconds is the same as 2 minutes. • (2) Used only for "relative time." EACH, the default, delays before each object; ONCE delays only the first object. • Uses Java's Thread.sleep() method and may not be exact in fractional seconds.
<i>devinfo</i>	Write Device Information <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>dfsrt</i>	Interface to DFSORT/CMS <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.
<i>diage4</i>	Submit Diagnose E4 Requests <ul style="list-style-type: none"> • Not implemented in Netrex Pipelines.

<i>dict hash</i>	<p>Read or Write a Dictionary</p> <p>NetRexx</p>  <ul style="list-style-type: none"> • Pipes for NetRexx only.
<i>dicta hasha</i>	<p>Write a Dictionary</p> <p>NetRexx</p>  <ul style="list-style-type: none"> • Pipes for NetRexx only.
<i>dictr hashr</i>	<p>Read a Dictionary</p> <p>NetRexx</p>  <ul style="list-style-type: none"> • Pipes for NetRexx only.
<i>dictw hashw</i>	<p>Write a Dictionary</p> <p>NetRexx</p>  <ul style="list-style-type: none"> • Pipes for NetRexx only.
<i>digest</i>	<p>Compute a Message Digest</p>  <p>(1) CMS Pipelines only. (2) NetRexx Pipelines only (dependent on the JVM implementation).</p> <p>NetRexx Pipelines returns the bytearray as a HEX string CMS returns a char array into the pipeline</p>
<i>disk file</i>	<p>Read a File</p>  <ul style="list-style-type: none"> • As in CMS, equivalent to diskr (Pipes for NetRexx Only) or <.

<div> <div>diska</div> <div>filea</div> <div>>></div> </div>	<div>Append to or Create a File</div> <div> </div>
<div> <div>diskback</div> <div>fileback</div> </div>	<div>Read a File Backwards</div> <div> <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines. </div>
<div> <div>diskfast</div> <div>filefast</div> </div>	<div>Read, Create, or Append to a File</div> <div> <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines. </div>
<div>diskid</div>	<div>Map CMS Reserved Minidisk</div> <div> <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines. </div>
<div> <div>diskr</div> <div>filer</div> <div><</div> </div>	<div>Read a File</div> <div> </div> <div> <ul style="list-style-type: none"> As in CMS, equivalent to diskr (Pipes for NetRexx Only) or <. </div>
<div> <div>diskrandom</div> <div>filerandom</div> </div>	<div>Random Access a File</div> <div> <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines. </div>
<div> <div>diskslow</div> <div>fileslow</div> </div>	<div>Read, Create, or Append to a File</div> <div> </div>
<div> <div>diskupdate</div> <div>fileupdate</div> </div>	<div>Replace Records in a File</div> <div> <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines. </div>
<div> <div>diskw</div> <div>filew</div> <div>></div> </div>	<div>Replace or Create a File</div> <div> </div>
<div> <div>display</div> <div>3.11</div> </div>	<div>Output to Web Browser</div>

NetRexx

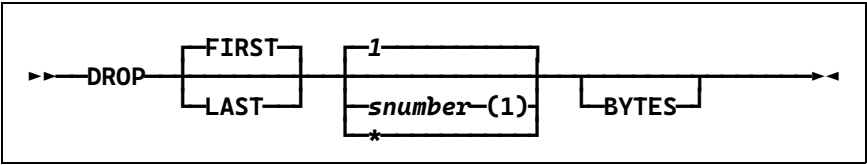
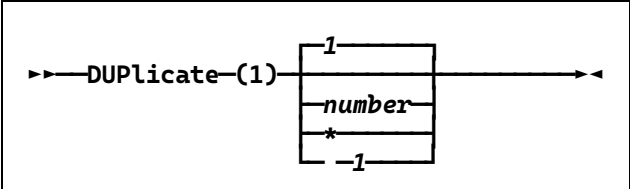
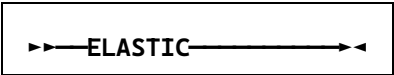
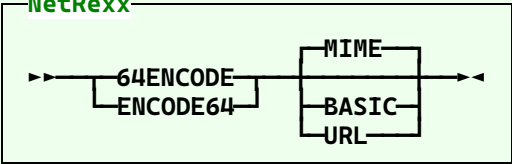
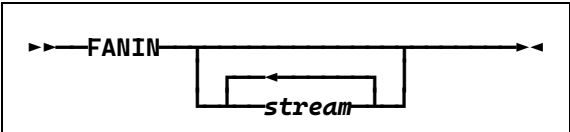


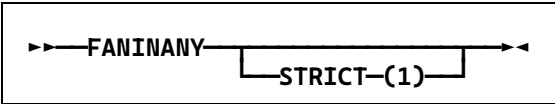
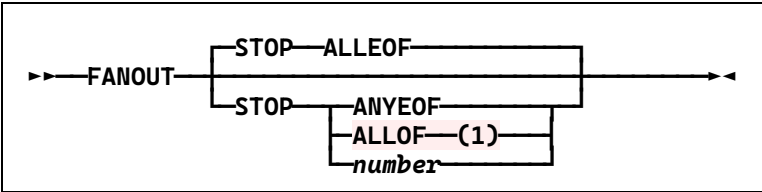
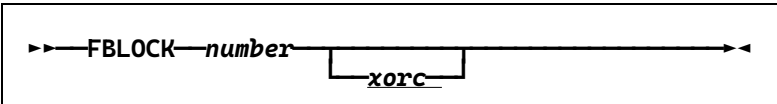
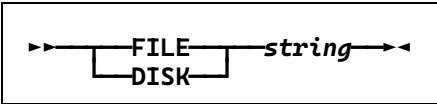
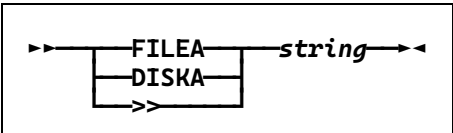
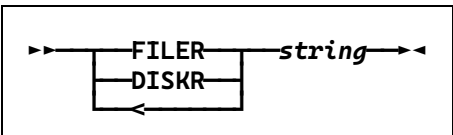
- DISPLAY works similar to and as a replacement for CONSOLE for output. But instead of going to the terminal window, it goes to a HTML file browser tab. This allows for HTML+CSS tags to control fonts, colors, and layout.
- To work, these are required outside Pipelines and NetRexx:
 - A working HTML browser program
 - The operating system to associate the filetype "html" with the browser, so the Pipelines stage "COMMAND PipeDisp.html" does call the browser and display the file.
 - The system have a Temp directory, known to Java.
- The DISPLAY stage overwrites the named file, by default PipeDisp.html, in the system Temp directory, then calls the COMMAND stage to display it. The file is not erased automatically by this stage.
- Each DISPLAY stage invocation opens a new browser tab, which remains open.
- The AS option causes the data to be surrounded by html tags.
 - The default TEXT or PRE puts on <pre> and </pre>. Most browsers use:
 - Fixed width font
 - Display all the white spaces: line feeds and multiple spaces
 - HTML uses <html> and </html>. Most browsers use:
 - Variable width font
 - Consolidate strings of white space into a single space
 - All the HTML tags
 - TABLE uses <table> and </table>
 - Expects the data records to begin with <tr><td> (or <tr><th>)
 - NOTAG uses <pre> & </pre>, but first converts all & characters to the entity & and < characters to < so HTML tags are not processed.
 - NONE uses no extra tags. Most browsers use:
 - HTML display
- OPTIONS QString is included in the opening tag for the AS option. This could be CLASS, STYLE, or other options.
- TITLE QString adds <title>delimitedString</title> to the beginning of the output. This should show as the title in the browser's tab.

Note: This officially should go into a HEAD section; here it won't be there. Most modern browsers will honor it anyplace in the file. If it is not honored as a tag, QString will be the top line of the display.
- STYLE QString adds <link rel="stylesheet" href="QString"> to the beginning of the output. This should include and use the named stylesheet. The name may have relative path names, or be an absolute file name. If there are spaces, enclose it in quotes.

Note: This officially should go into a HEAD section; here it won't be there. Most modern browsers will honor it anyplace in the file. If it is not honored as a tag, it will not show -- except in the NOTAG option. The file itself is copied from its stated location into the system Temp directory, overwriting any existing file. This file is not erased automatically by this stage.
- FILETYPE may be used to change the default "html". This permits use of other types that MAY be preprocessed if the system, external to Pipelines, is set up to recognize it, for example, "JSP" or "PHP". A "dot" is optional; only one will be used.

Note: filetypes other than .html may be handled by the system by some program other than the browser.
- QString: It is optional to enclose the name in quotes, but quotes are required if the name includes spaces.
- FILENAME may be used to write and display another file. It may include a path designation, either absolute or relative. A relative path is based on the working directory. If no path is specified in the name, the system Temp

<p><i>drop</i></p>	<p>Discard Records from the Beginning or the End of the File</p>  <ul style="list-style-type: none"> (1) CMS: must be positive. NetRexx Pipelines: negative reverses FIRST/LAST, so DROP FIRST -3 is the same as DROP LAST 3.
<p><i>duplicate</i> <i>duplicat</i> <i>duplica</i> <i>duplic</i> <i>dupli</i> <i>dupl</i> <i>dup</i></p>	<p>Copy Records</p>  <ul style="list-style-type: none"> (1) CMS is DUPLICAT due to 8-character name limitation
<p><i>elastic</i></p>	<p>Buffer Sufficient Records to Prevent Stall</p> 
<p><i>encode64</i> <i>64encode</i> <i>3.11</i></p>	<p>Encode to Base-64 Format</p> <p>NetRexx</p>  <ul style="list-style-type: none"> NOTE: CMS is only 64DECODE, and does not have the options; it does MIME. BASIC - Output is mapped to a set of characters lying in A-Za-z0-9+/. The encoder does not add any line feed in output, and the decoder rejects any character other than A-Za-z0-9+/. URL - Output is mapped to set of characters lying in A-Za-z0-9+_. Output is URL and filename safe. MIME - Output is mapped to MIME friendly format. Output is represented in lines of no more than 76 characters each, and uses a carriage return '\r' followed by a linefeed '\n' as the line separator. No line separator is present to the end of the encoded output. 3.11: New to NetRexx. Add MIME, BASIC, & URL options.
<p><i>eofback</i></p>	<p>Run an Output Device Driver and Propagate End-of-File Backwards</p> <ul style="list-style-type: none"> Not implemented in Netrex Pipelines.
<p><i>escape</i></p>	<p>Insert Escape Characters in the Record</p> <ul style="list-style-type: none"> Not implemented in Netrex Pipelines.
<p><i>fanin</i></p>	<p>Concatenate Streams</p> 

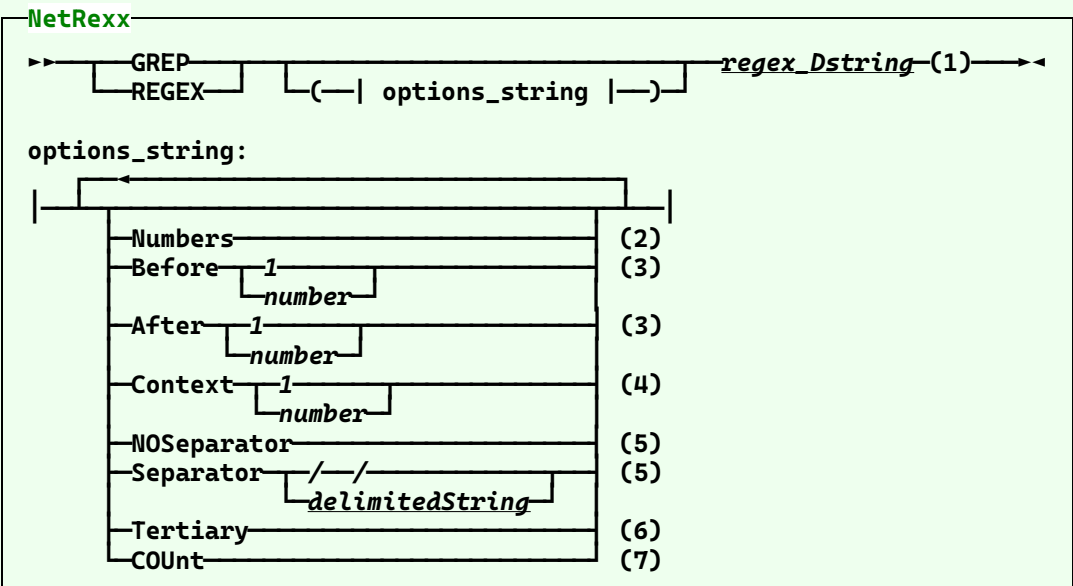
<i>faninany</i>	<p>Copy Records from Whichever Input Stream Has One</p>  <ul style="list-style-type: none"> • (1) CMS only.
<i>fanintwo</i>	<p>Pass Records to Primary Output Stream</p>
<i>fanout</i>	<p>Copy Records from the Primary Input Stream to All Output Streams</p>  <ul style="list-style-type: none"> • (1) CMS only
<i>fanouttwo</i>	<p>Copy Records from the Primary Input Stream to Both Output Streams</p>
<i>fbaread</i>	<p>Read Blocks from a Fixed Block Architecture Drive</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>fbawrite</i>	<p>Write Blocks to a Fixed Block Architecture Drive</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>fblock</i>	<p>Block Data, Spanning Input Records</p> 
<i>file disk</i>	<p>Read or Write a File</p> 
<i>filea diska >></i>	<p>Append to or Create a File</p> 
<i>fileback diskback</i>	<p>Read a CMS file backwards</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>filedescriptor</i>	<p>Read or Write an OpenExtensions File that Is Already Open</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>filefast diskfast</i>	<p>Read or write a CMS file</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>filer file disk diskr <</i>	<p>Read a File</p> 
<i>filerandom diskrandom</i>	<p>Read specific records from a CMS file</p>

<i>fileslow</i> <i>diskslow</i>	Read, Create, or Append to a File <div> </div>
<i>filetoken</i>	Read or Write an SFS File That is Already Open <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>fileupdate</i> <i>diskupdate</i>	Change records in a CMS file
<i>file</i> <i>diskw</i> >	Replace or Create a File <div> </div>
<i>fillup</i>	Pass Records To Output Streams <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>filterpack</i>	Manage Filter Packages <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>find</i>	Select Lines by XEDIT Find Logic <div> </div>
<i>fitting</i>	Source or Sink for Copipe Data <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>fmtfst</i>	Format a File Status Table (FST) Entry <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>frlabel</i> <i>fromlabel</i>	Select Records from the First One with Leading String <div> </div>
<i>fromlabel</i> <i>frlabel</i>	Select Records from the First One with Leading String <div> </div>
<i>frtarget</i>	Select Records from the First One Selected by Argument Stage <div> </div>
<i>fullscreen</i> <i>fullscree</i> <i>fullscre</i> <i>fullscr</i>	Full screen 3270 Write and Read to the Console or Dialed/Attached Screen <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>fullscrq</i>	Write 3270 Device Characteristics <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>fullscrq</i>	Write 3270 Device Characteristics <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>fullscrs</i>	Format 3270 Device Characteristics <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.

<i>gate</i>	Pass Records Until Stopped <div> </div>
<i>gather</i>	Copy Records From Input Streams <ul style="list-style-type: none"> Not implemented in Netrex Pipelines.
<i>gen</i>	Generate a Sequence of Numbers Starting with 1 <div> </div> <ul style="list-style-type: none"> Not implemented in CMS Pipelines.
<i>getfiles</i> <i>getfiles</i> <i>getfile</i> <i>getfil</i> <i>getfi</i> <i>getf</i> <i>get</i>	Read Files <div> </div>
<i>getovers</i>	Write the Contents of Objects <div> </div> <ul style="list-style-type: none"> Input stream 0 should contain rexx objects. The getovers stage will output the index and contents of the stem on stream 0. If output stream 1 is connected, the root is placed there. Any severed streams will cause then stage to exit. Passing a non rexx object will cause the stage to exit with return code 13. Pipes for NetRexx only.
<i>getstems</i>	Write the Contents of Members of Stems <div> </div> <ul style="list-style-type: none"> Input stream 0 should contain rexx objects containing stems. The getstems stage will output the contents of the stem on stream 0. If output stream 1 is connected, the root is placed there. Any severed streams will cause then stage to exit. Passing a non rexx stem object will cause the stage to exit with return code 13. Pipes for NetRexx only.

grep
regex
3.09

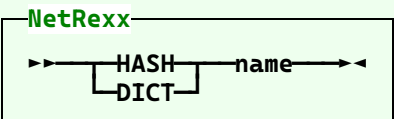
Select Lines by a Regular Expression



- NetRexx Pipelines only.
- Records matching the RegEx are put out on primary output.
- Records not matching are put out on secondary, if connected, or discarded.
- .
- (1) Regex_string is a Java RegEx expresion. Null string passes all records.
- (2) Records are prefaced with records number, 10 characters, right justified.
- (3) Number of records put out after a matching record.
- (4) Number of records put out before and after a matching record.
- (5) Inserted before a group of "before records" or the found record with "after records."
- (6) Send all matching records (no numbers) to tertiary output stream, if connected.
- (7) Only a count of matches is put out on the primary output stream. (Other options probably should not be used with this.)

hash
dict

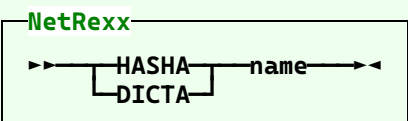
Read or Write a Dictionary



- Pipes for NetRexx only.

hasha
dicta

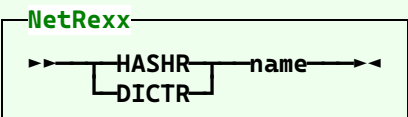
Write a Dictionary



- Pipes for NetRexx only.

hashr
dictr

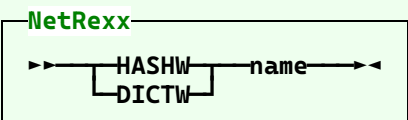
Read a Dictionary



- Pipes for NetRexx only.

hashw
dictw

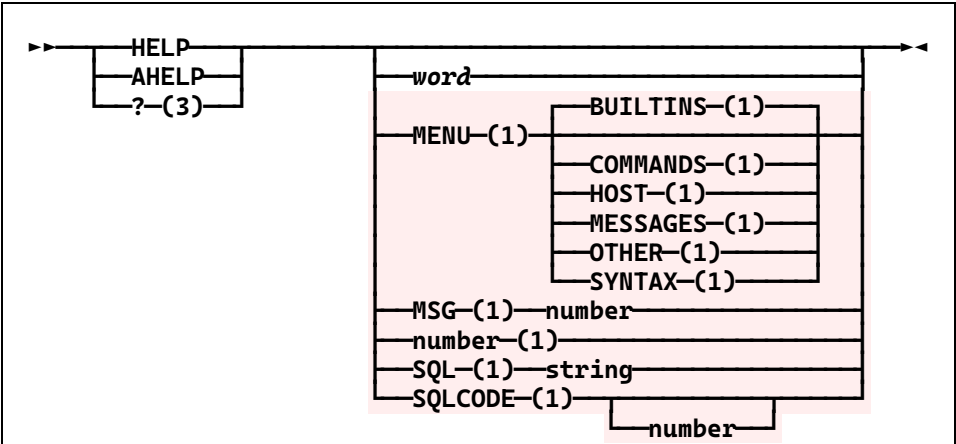
Write a Dictionary



- Pipes for NetRexx only.

help
ahelp
?

Display Help for Pipelines



- (1) CMS Pipelines only. Not yet in NetRexx Pipelines.
- (2) If primary output is connected, lines are propagated, otherwise they are sent to the console by "say."
- (3) ? is the default pipeEnd character. Here it is useful only when a different pipeEnd is defined.

hfs
bfs

Read or Append File in the Hierarchical File System

- Not implemented in Netrex Pipelines.

hfsdirectory
hfsdir
bfsdirectory
bfmdir

Read Contents of a Directory in a Hierarchical File System

- Not implemented in Netrex Pipelines.

hfsquery
hfsq
bfsquery
bfsq

Write Information Obtained from OpenExtensions into the Pipeline

- Not implemented in Netrex Pipelines.

hfsreplace
hfsrep
bfsreplace
bfrep

Replace the Contents of a File in the Hierarchical File System

- Not implemented in Netrex Pipelines.

hfsstate
hfsstat
bfsstate
bfstat

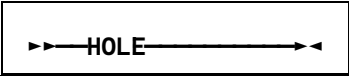

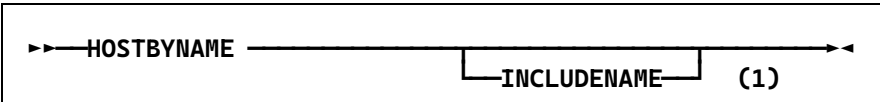
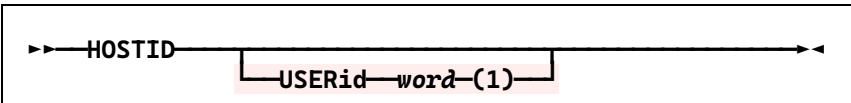
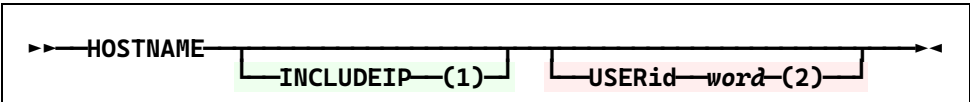
Obtain Information about Files in the Hierarchical File System

- Not implemented in Netrex Pipelines.

hfsxecute
hfsx
bfsxecute
bfxx

Issue OpenExtensions Requests

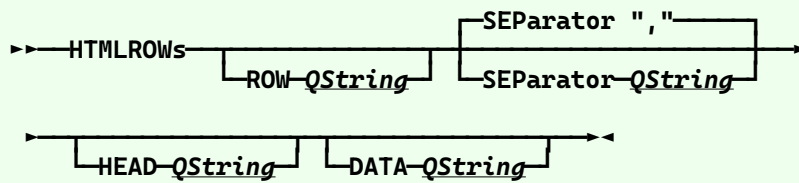
- Not implemented in Netrex Pipelines.

<i>hlasm</i>	Interface to High Level Assembler <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>hlasmerr</i>	Extract Assembler Error Messages from the SYSADATA File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>hole</i>	Destroy Data <div>  </div>
<i>hostbyaddr</i> 3.09	Resolve IP Address into Domain and Host Name <div>  </div> <ul style="list-style-type: none"> (1) Optional parameter not present in VM/CMS version INCLUDEIP - Also include the IP address along with the hostname. Output: <hostname>/<ip address> Example: <code>dns.google/8.8.8.8</code> Known issues: The underlying Java method getByName/getHostName does not appear to handle IPv6 addresses in any known and consistent manner. Could be related to a host configuration issue but googling shows odd and inconsistent results for getting around this.
<i>hostbyname</i> 3.09	Resolve a Domain Name into an IP Address <div>  </div> <ul style="list-style-type: none"> (1) Optional parameter not present in CMS Pipelines Arguments: INCLUDENAME - Also include the name of the host on output. Output: <hostname>/<ip address> Example: <code>dns.google/8.8.8.8</code>
<i>hostid</i> 3.09	Write TCP/IP Default IP Address <div>  </div> <ul style="list-style-type: none"> (1) The USERid option available under CMS Pipelines is not applicable and is ignored in NetRexx Pipelines
<i>hostname</i> 3.09	Write TCP/IP Host Name <div>  </div> <ul style="list-style-type: none"> (1) Optional parameter not present in VM/CMS version (2) The USERid option available under CMS is not applicable and is ignored in NetRexx Pipelines Arguments: INCLUDEIP - include the IP address of the system in the response in the form <hostname>/<ip address>

htmlrows
htmlrow
3.11

Convert rows to HTML format

NetRexx



- HTMLROWS reads rows from its primary input stream and writes them to its primary output stream, altering them to have the proper HTML tags for TABLE ROWS.
- I.e., it converts
abc,mnop,xyz
into
<tr><td>abc</td><td>mnop</td><td>xyz</td></tr>
- The SEPARATOR QString, by default the comma character, can be specified.
- There are options to put additional data inside the tags. This could be used for class or style tag options, for example.
 - ROW QString : puts its information into the <tr>-tags
 - DATA QString : puts its information into the <td>-tags
 - HEAD QString : puts its information into the <th>-tags (1)
- QString is a quoted string of characters. The quote marks may be either single or double, but must match. If there are no spaces in the string, the quote marks are optional.
- (1) If there is a HEAD option, the first row read has <th>-tags instead of <td>-tags. It must have a QString of at least "". Succeeding rows have the standard <td>-tags.

httpsplit

Split HTTP Data Stream

- Not implemented in Netrex Pipelines.

iebcopy

Process IEBCOPY Data Format

- Not implemented in Netrex Pipelines.

if

Process Records Conditionally

- Not implemented in Netrex Pipelines.

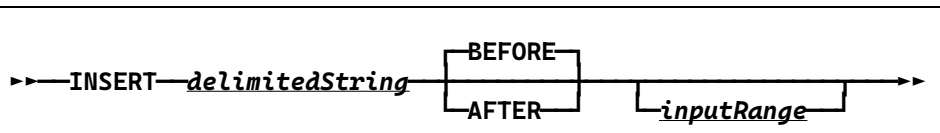
immcmd

Write the Argument String from Immediate Commands

- Not implemented in Netrex Pipelines.

insert

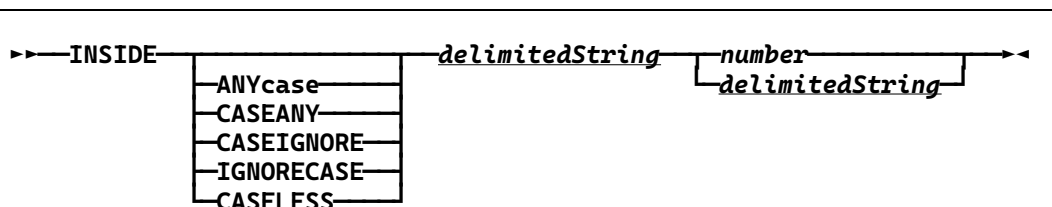
Insert String in Records



- insert a string into a record before or after the record content. Will be much more efficient than specs especially if the input is a Byte[]

inside

Select Records between Labels

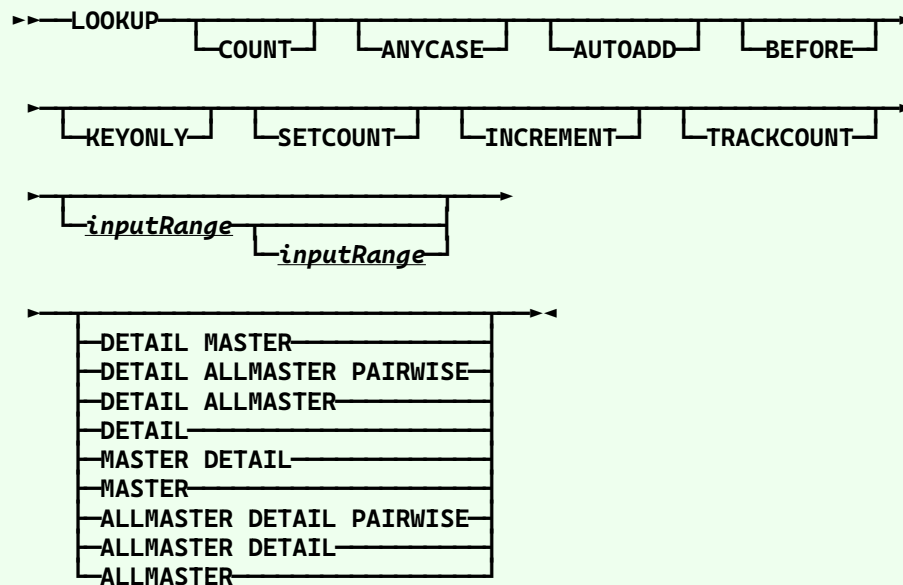


<i>instore</i>	Load the File into a storage Buffer <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>ip2socka</i>	Build sockaddr_in Structure <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>ispf</i>	Access ISPF Tables <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>jeremy</i>	Write Pipeline Status to the Pipeline <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>join</i>	Combine Records <div> </div>
<i>joincont</i>	Join Continuation Lines <div> </div>
<i>juxtapose</i>	Preface Record with Marker <div> </div>
<i>ldrtbls</i>	Resolve a Name from the CMS Loader Tables <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>listcat</i>	Obtain Data Set Names <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>listdsi</i>	Obtain Information about Data Sets <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>listispf</i>	Read Directory of a Partitioned Data Set into the Pipeline <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.

<i>listpds</i>	<div>Read Directory of a Partitioned Data Set into the Pipeline</div> <div><ul style="list-style-type: none">Not implemented in Netrex Pipelines.</div>
<i>listzip</i>	<div>List the Files in a Zipped File</div> <div><div>NetRexx</div><div>►►LISTZIPzipFileName◄◄</div></div>
<i>literal</i>	<div>Write the Argument String</div> <div><div>►►LITERAL◄◄</div><div>string</div></div>
<i>locate</i> <i>locat</i> <i>loca</i> <i>loc</i> <i>lo</i> <i>l</i>	<div>Select Lines that Contain a String</div> <div><div>►►Locate</div><div><div>ANYCase</div><div>CASEANY</div><div>CASEIGNORE</div><div>IGNORECASE</div><div>CASELESS</div></div><div><div>MIXED(1)</div><div>ONEs(1)</div><div>ZEROs(1)</div></div><div>inputRanges</div><div>ANYof</div></div> <div><div>delimitedString</div></div> <div><div>(1) Not in NetRexx Pipelines, yet.</div><div><ul style="list-style-type: none">[2] IBM documentation has this as "LOCATE" rather than "Locate". But the abbreviations work in both systems.</div></div>

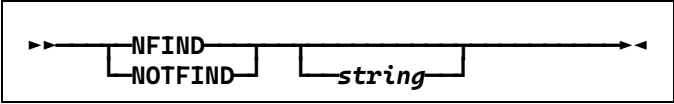

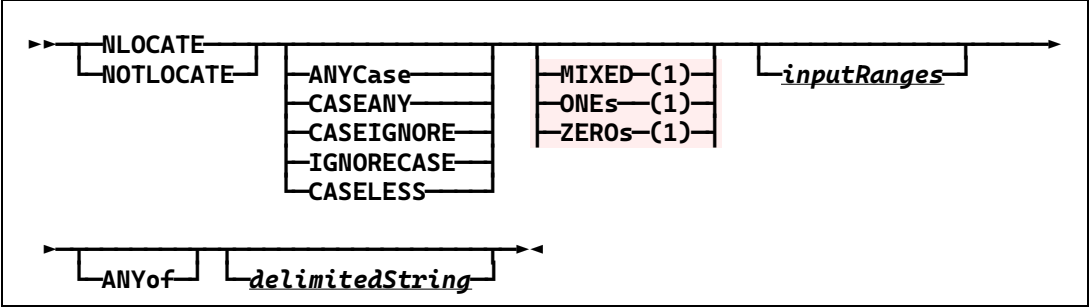

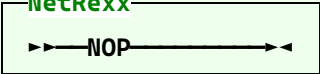
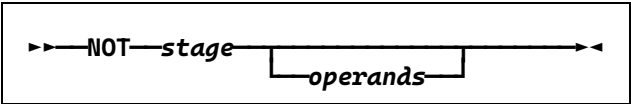
Find Records in a Reference Using a Key Field

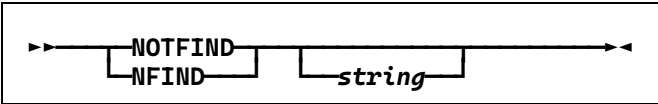
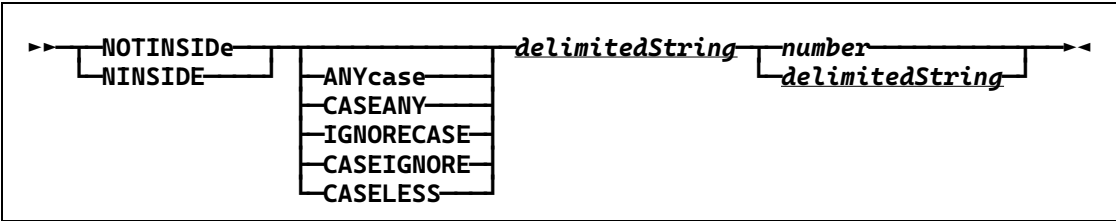
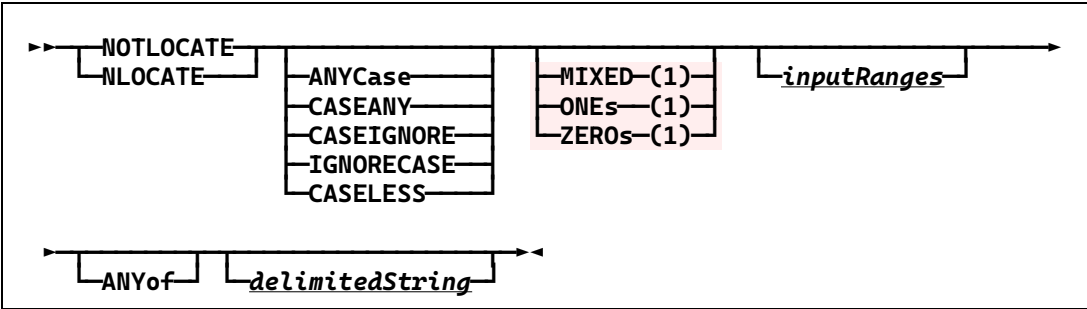
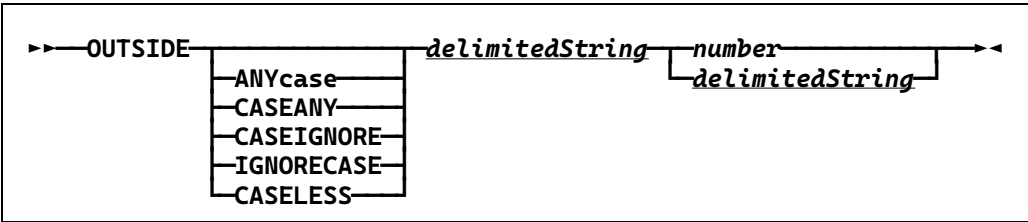
NetRexx



- in stream 0 are detail records
- in stream 1 are master records
- in stream 2 adds to masters
- in stream 3 delete from masters
-
- out stream 0 are matched records
- out stream 1 are unmatched detail records
- out stream 2 are unmatched or counted master records
- out stream 3 deleted masters
- out stream 4 duplicate masters
- out stream 5 unmatched master deletes
-
- lookup does not consider an unconnected output stream an error. It does propagate EOFs from output streams.

<i>lookup</i>	<p>Find Records in a Reference Using a Key Field</p> <p>CMS</p> <ul style="list-style-type: none"> • in stream 0 are detail records • in stream 1 are master records • in stream 2 adds to masters • in stream 3 delete from masters • • out stream 0 are matched records • out stream 1 are unmatched detail records • out stream 2 are unmatched or counted master records • out stream 3 deleted masters • out stream 4 duplicate masters • out stream 5 unmatched master deletes • • lookup does not consider an unconnected output stream an error. It does propagate EOFs from output streams.
<i>maclib</i>	<p>Generate a Macro Library from Stacked Members in a COPY File</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>mapmdisk</i>	<p>Map Minidisks Into Data spaces</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>mctoasa</i>	<p>Convert CCW Operation Codes to ASA Carriage Control</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>mdiskblk</i>	<p>Read or Write Minidisk Blocks</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>mdskrandom</i> <i>mdskrand</i>	<p>Random Access a CMS File on a Mode</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.
<i>mdskslow</i>	<p>Read, Append to, or Create a CMS File on a Mode</p> <ul style="list-style-type: none"> • Not implemented in Netrexx Pipelines.

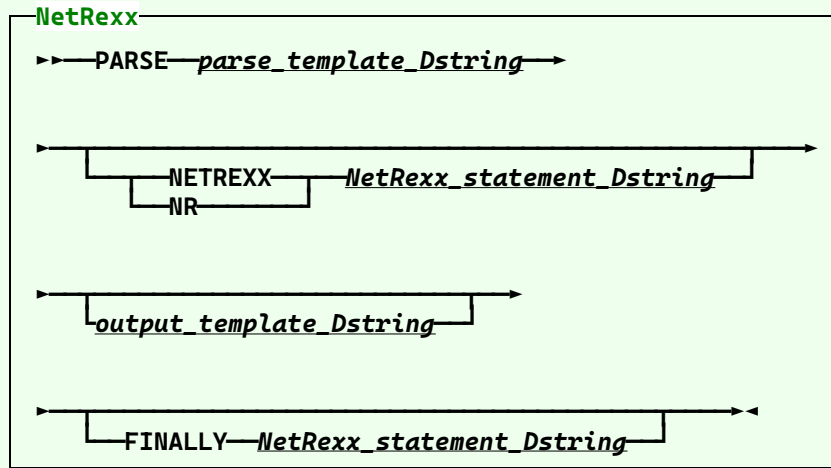
<i>mdskupdate</i> <i>mdskupda</i>	Replace Records in a File on a Mode <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>members</i> <i>member</i>	Extract Members from a Partitioned Data Set <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>merge</i>	Merge Streams <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>mqsc</i>	Issue Commands to a WebSphere MQ Queue Manager <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>nfind</i> <i>notfind</i>	Select Lines by XEDIT NFind Logic 
<i>ninside</i> <i>notinsid</i> <i>notinside</i> 3.09	Select Records Not between Labels 
<i>nlocate</i> <i>notlocate</i>	Select Lines that Do Not Contain a String  <ul style="list-style-type: none"> (1) Not in NetRexx Pipelines, yet.
<i>noEofBack</i>	Pass Records and Ignore End-of-file on Output 
<i>nop</i>	No Operation <p>NetRexx</p>  <ul style="list-style-type: none"> Pipes for NetRexx only.
<i>not</i>	Run Stage with Output Streams Inverted 

<i>notfind</i> <i>nfind</i>	Select Lines by XEDIT NFind Logic 
<i>notinside</i> <i>notinsid</i> <i>ninside</i>	Select Records Not between Labels 
<i>notlocate</i> <i>nlocate</i>	Select Lines that Do Not Contain a String  <ul style="list-style-type: none"> • (1) Not in NetRexx Pipelines, yet.
<i>nucext</i>	Call a Nucleus Extension <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>optcdj</i>	Generate Table Reference Character (TRC) <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>outside</i>	Select Records Not between Labels 
<i>outstore</i>	Unload a File from a storage Buffer <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>over</i>	Write the Values of Stems <ul style="list-style-type: none"> • Obsolete. Now use varover. over is now an alias for overlay..

<div>overlay</div> <div>overla</div> <div>overl</div> <div>over</div>	<div>Overlay Data from Input Streams</div> <div> </div> <div> <ul style="list-style-type: none"> • HOLD keeps the last record from each stream, except primary, and uses it if the stream ends. • TRANSPARENT means that character can be different from the PAD character. If omitted, it is the same as PAD character. • dstream can be used instead of a non-primary stream. • (1) NetRexx Pipelines only • (2) same as highest (+1) stream; implies HOLD </div>
<div>overstr</div>	<div>Process Overstruck Lines</div> <div> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines. </div>
<div>pack</div>	<div>Pack Records as Done by XEDIT and COPYFILE</div> <div> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines. </div>
<div>pad</div>	<div>Expand Short Records</div> <div> </div>
<div>parcel</div>	<div>Parcel Input Stream Into Records</div> <div> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines. </div>

parse
3.09
4.05

Rearrange Contents of Records



- Records are parsed via the parse_template_delimited_string.
- Variables are named `_n`, where `n` is 1 to 9.
- The values of the variables are put into the output_template_delimited_string replacing `_n`.
- For a literal `_n` that won't be changed, use `__n`.
- The two NetRexx_statement_Dstrings are single statements, or multiple statements separated by ";"s.
 - The `_n` variables can be used and changed.
 - The string `\n` will split the string into separate output records.
 - The special indexed REXX variable COUNTER[] is also available in these Dstrings. This is specific to a PARSE stage, but persists between records. All the indexed values are initiated to 0. Both indexes and values can be strings.
 - This is powerful and has the possibility of doing damage to your pipe.
You have been warned!
- If the NR NetRexx_statement_Dstring returns a value, it is used as the output instead of the optional output_template_Dstring.
- The FINALLY's statement_Dstring is executed after the last input record has been processed. The value returned is put out as an "extra" output record.
- (As of 4.05) Variable names of "\$n" are depreciated, and can not be used with NETREXX or FINALLY options.
- NetRexx Pipelines only.
- .
- Examples:
 - `parse / 2 _1 +1/ /The second letter is "_1". __1 won't be changed./`
 - `parse /2 _1 +1/ NR /counter[1]=counter[1]+1; _9=counter[1]/ /_9/
FINALLY /return "Count:" _9/`
 - `PARSE / _1 2 _2 +1 _3/ ,
NR /if _2.datatype('L') then counter['c'] = counter['c'] + 1; _2 = _2.upper/ ,
/_1_2_3/ ,
FINALLY /return counter['c'] 'Changed to upper'/`

pause

Signal a Pause Event

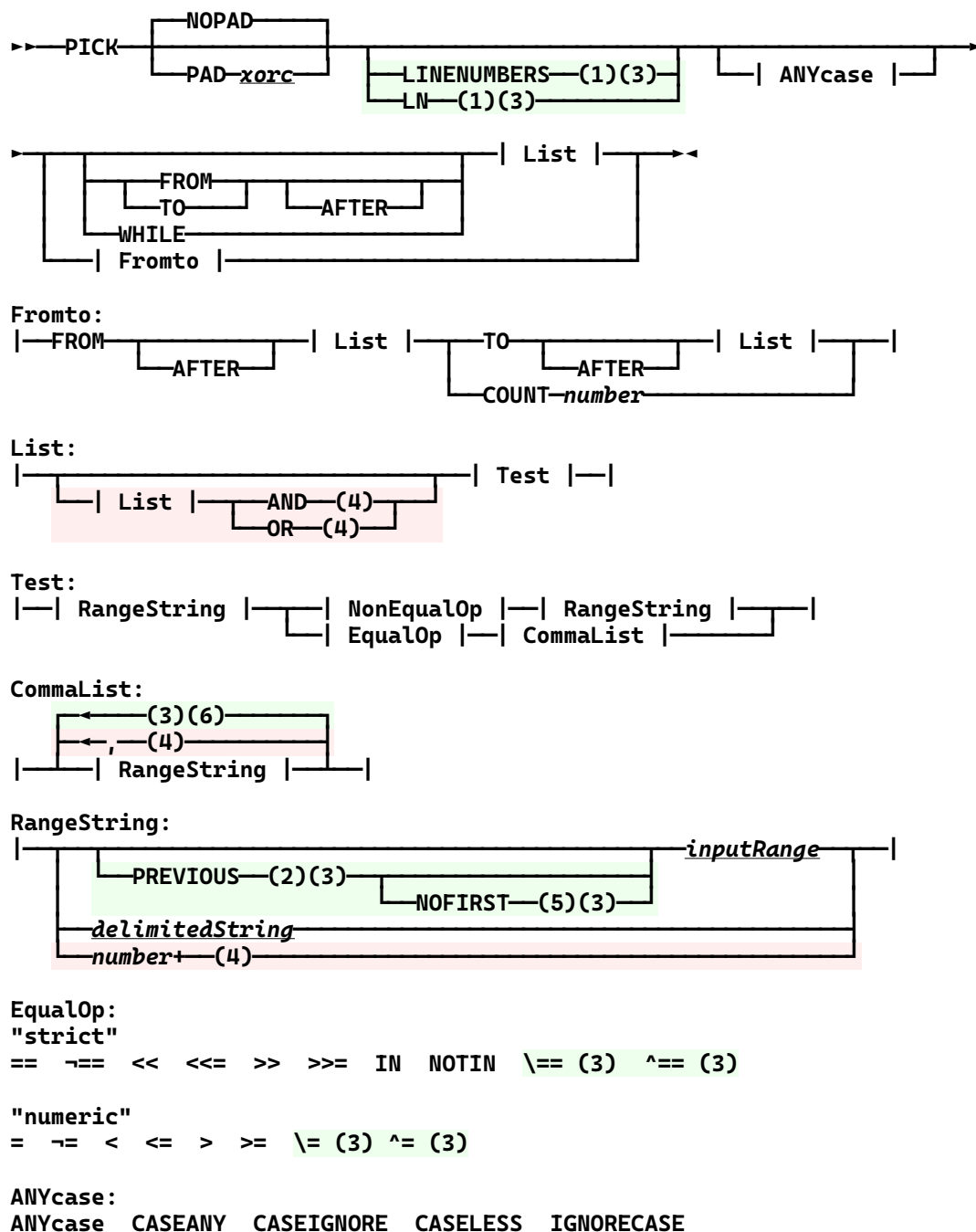
- Not implemented in Netrex Pipelines.

pdsdirect
pds

Write Directory Information from a CMS Simulated Partitioned Data Set

- Not implemented in Netrex Pipelines.

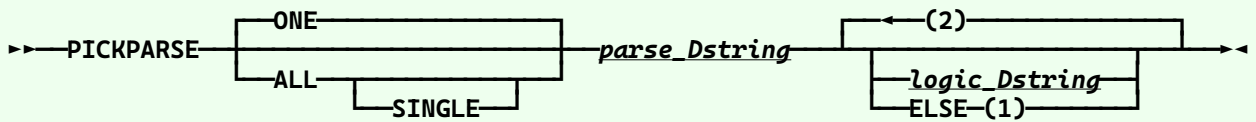
Select Lines that Satisfy a Relation



- (1) NetRexx only. Inserts the original record number followed by a SPACE at the beginning of each output record.
 - (2) NetRexx only. Uses the data from the previous record. Before the first record, this is Rexx "".
 - (3) NetRexx Pipelines only. Not yet in CMS Pipelines.
 - (4) CMS Pipelines only. Not yet in NetRexx Pipelines.
 - (5) NetRexx Only. Uses first record data for first record instead of previous "".
 - (6) CMS uses ",", NetRexx does not. CMS limits RangeStrings to right side, NetRexx allows them on the left, too.
- CMS also allows only == or <== with RangeStrings. NetRexx permits any comparison op. NetRexx concatenates the several ranges for comparison.

Select Lines that Satisfy Relations using REXX Parse

NetRexx



- Records are parsed via the `parse_delimited_string`.
- Variables are named `$n`, where `n` is 1 to 9.
- The values of the variables are put into the `logic_delimited_string` replacing `$n` and evaluated. If TRUE, the record is put out on the stream numbered by the `dstring`'s position.
- The stream for a `Dstring` of ELSE is used if no previous logic `Dstring` is TRUE.
- If there is no specific ELSE, there is an implied one at the end; if that stream is not connected, the record is discarded.
- If ONE then the record is put out on, at most, one stream: the first one matched.
- If ALL then the record is put out on all streams matched.
- If SINGLE then the records are all put out on the primary output stream.
- The `parse_delimited_string` and `logic_delimited_string(s)` follow normal NetRexx rules.
- (1) Implied ELSE after last specified `dstring`.
- (2) Up to 10 logic `Dstrings` may be specified to go to up to 11 output streams (including an implied ELSE).
- Not implemented in CMS Pipelines.

Pickparse permits selecting records by a NetRexx logical expression, using parts of the record selected by a REXX PARSE template.

A simple example has two delimited strings, a REXX template and a logical expression:

```
pickparse / . . $3 . 50 $5 +5 / / $3 < $5 /
```

The parse template selects the 3rd word, and the 5 characters starting in column 50. the variable names are a dollar sign and a digit. Then those variables can be used in the logic expression. When run, and records matching the logic expression are written to the primary output stream, others to the secondary. If either stream is not connected, the corresponding records are discarded.

There can be multiple logic expressions, each in its own delimited string. Parenthetical expressions may be used. Records are matched to each in turn. Any records matching are written to that output stream, if connected.

With the option ONE, the default, each record is written to one output stream: the first one it matches. With the option ALL, the matching goes on and a record could be written to multiple output streams.

There is an implicit or explicit ELSE as the last logic expression. Records that have not matched any of the previous expressions match this and are written or discarded depending on if the stream is connected or not.

The parse template can define up to 9 separate zones, \$1 to \$9. The variables `$_n` are also available for the logic expressions; they are the values from the previous record. Initially these are "".

There can be up to 10 output streams defined, and up to 9 logic expressions plus ELSE.

Issue Pipeline Commands

- Not implemented in Netrex Pipelines.

Terminate Stages Waiting for an External Event

- Not implemented in Netrex Pipelines.

Reverse Polish Expression Parser

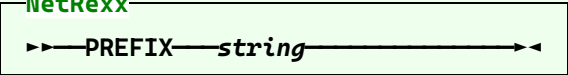
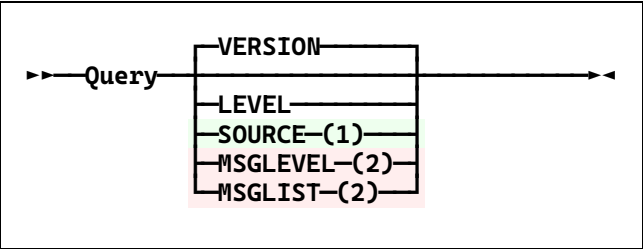
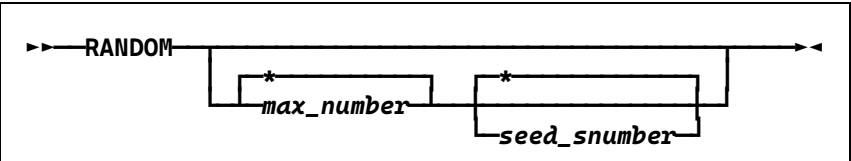
- Not implemented in Netrex Pipelines.

Control Destructive Test of Records

- Not implemented in Netrex Pipelines.

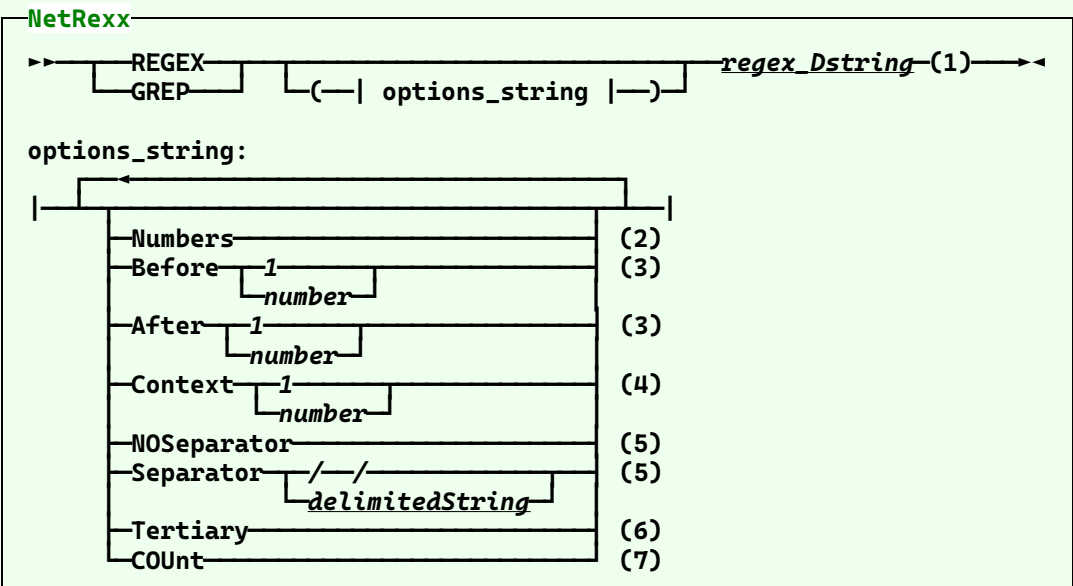
Put Output from a Device Driver before Data on the Primary Input Stream

- Not implemented in Netrex Pipelines.

<i>prefix</i>	<p>Stop and Run a Stage First, Before Continuing</p> <p>NetRexx</p>  <ul style="list-style-type: none"> • Blocks its primary input and excutes stage supplied as an argument. The output from this stage are put to the primary output stream. When its compete the primary input is shorted. • Not implemented in CMS Pipelines.
<i>printmc</i>	<p>Print Lines</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>punch</i>	<p>Punch Cards</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>qpdecode</i>	<p>Decode to Quoted-printable Format</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>qpencode</i>	<p>Encode to Quoted-printable Format</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>qsam</i>	<p>Read or Write Physical Sequential Data Set through a DCB</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>query</i>	<p>Obtain Information From Pipelines</p>  <ul style="list-style-type: none"> • (1) Not CMS • (2) Not NetRexx Pipelines
<i>random</i> <i>3.09</i>	<p>Generate Pseudorandom Numbers</p>  <ul style="list-style-type: none"> • NetRexx Pipelines will be a different sequence than CMS gives with the same seed.
<i>reader</i>	<p>Read from a Virtual Card Reader</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.
<i>readpds</i>	<p>Read Members from a Partitioned Data Set</p> <ul style="list-style-type: none"> • Not implemented in NetrexX Pipelines.

regex
grep
3.09

Select Lines by a Regular Expression



- NetRexx Pipelines only.
- Records matching the RegEx are put out on primary output.
- Records not matching are put out on secondary, if connected, or discarded.
- .
- (1) Regex_string is a Java RegEx expresion. Null string passes all records.
- (2) Records are prefaced with records number, 10 characters, right justified.
- (3) Number of records put out after a matching record.
- (4) Number of records put out before and after a matching record.
- (5) Inserted before a group of "before records" or the found record with "after records."
- (6) Send all matching records (no numbers) to tertiary output stream, if connected.
- (7) Only a count of matches is put out on the primary output stream. (Other options probably should not be used with this.)

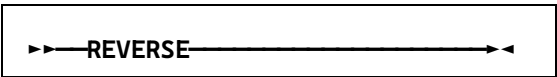
retab

Replace Runs of Blanks with Tabulate Characters

- Not implemented in NetrexX Pipelines.

reverse

Reverse Contents of Records



rexx

Run a REXX Program to Process Data

- Not implemented in NetrexX Pipelines.

rexxvars

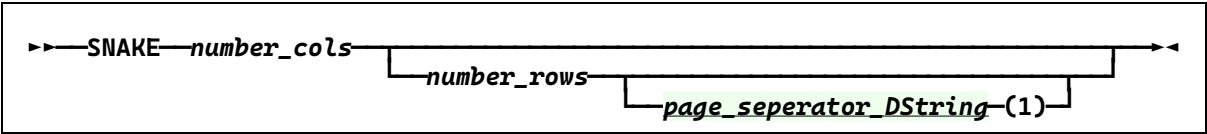
Retrieve Variables from a REXX or CLIST Variable Pool

- Not implemented in NetrexX Pipelines.

<i>runpipe</i>	Issue Pipelines, Intercepting Messages <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>scm</i>	Align REXX Comments <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>sec2greg</i>	Convert Seconds Since Epoch to Gregorian Timestamp <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>select</i> 4.06 5.01	Select Records using user logic <div> NetRexx </div> <ul style="list-style-type: none"> Using the default TRUEfalse, or TF, option, records are selected by evaluating the NetRexx <i>T/F_Delimited_string</i>. The Dstring is placed in a method that supplies the record in the variable <i>rec</i>. The previous record is in <i>prev</i>. The method returns 1 to select the record to the primary output stream, or 0 to send it to the secondary stream. Using the NOT option reverses the logic, and unmatched records are sent to the primary output stream. Alternatively, using the MULTIPLE or NOT option, with a <i>Digit_NetRexx_Dstring</i>, which evaluates to a 0 to 9 digit, the method can return the number of any output stream. Again, the record in the variable <i>rec</i>, previous record is in <i>prev</i>. (Note: failure to use the MULTIPLE option with multiple outputs will reverse the primary output stream, 0, and the secondary, 1.) Any other return value results in the record being discarded. Any record sent to a disconnected stream is discarded. Caution: Since any NetRexx statement can be used, this is powerful and could cause problems. Also, due to the late compiling, at stage run time, debugging can be difficult. The reported line numbers have nothing to do with your code. 5.01 adds the four options. <p>Examples:</p> <pre>select true /return rec.pos('2') > 0/ select multiple /parse rec 2 r +1;parse prev 2 p +1; return r \= p/</pre>
<i>serialize</i>	Convert Objects to/from a Single Text String <div> NetRexx </div> <ul style="list-style-type: none"> (1) classname if class is specified deserialize input to objects of this type - otherwise serialize input objects. (2) Pipes for NetRexx only. (3) For some reason readObject does not like more than one object network in its stream. Block multiple objects. (4) See examples/serialize_tests01.njp
<i>sfsback</i>	Read an SFS File Backwards <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>sfsdirectory</i>	List Files in an SFS Directory <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>sfsrandom</i>	Random Access an SFS File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.
<i>sfsupdate</i>	Replace Records in an SFS File <ul style="list-style-type: none"> Not implemented in Netrexx Pipelines.

snake
3.09

Build Multicolumn Page Layout



- (1) NetRexx Pipelines only. Appears first, last, and between pages.
Avoid \ as escape terms maybe added in the future. \n for newline is OK.
Your system may require \\n .

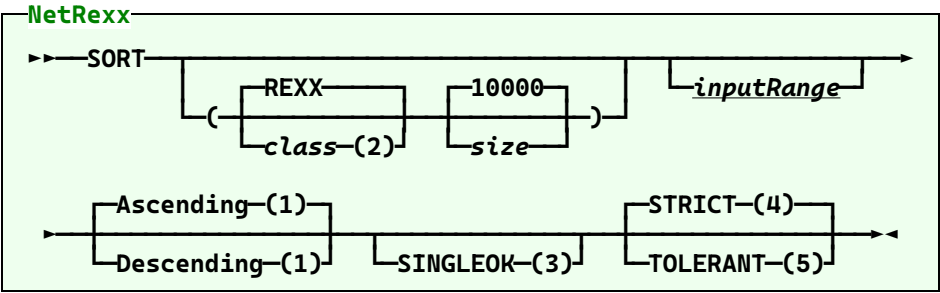
socka2ip

Format sockaddr_in Structure

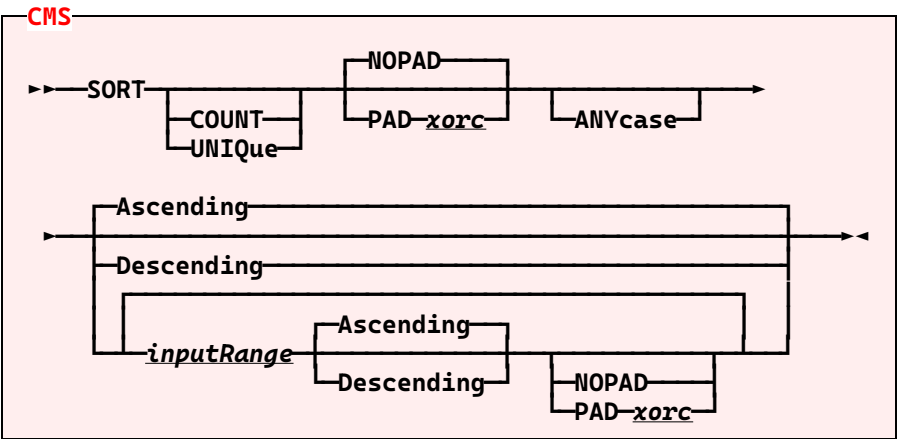
- Not implemented in Netrex pipelines.

sort

Order Records

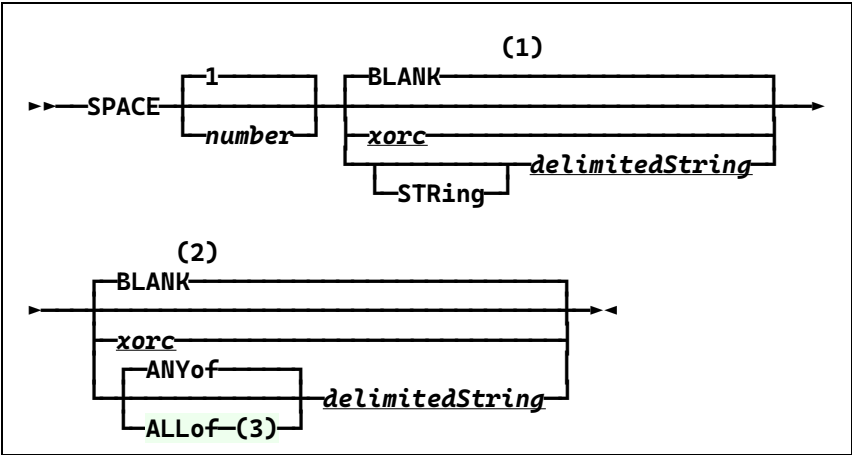


- (1) May come before inputRange, for backwards compatibility.
- (2) Requires that you implement another sortClass with a name beginning with 'sort'
- (3) Suppresses error message if only one record to sort for Rexx objects.
- (4) Uses strict Rexx comparisons rather than tolerant.
- (5) Uses tolerant Rexx comparisons rather than strict.
 - Numerics are compared numerically.
 - Strings are compared caseless.
 - Leading blanks are ignored.
- Uses sortClass class as Interface Class for Generic Sort Objects and sortRexx class to Sort Rexx Text Objects



space
3.09

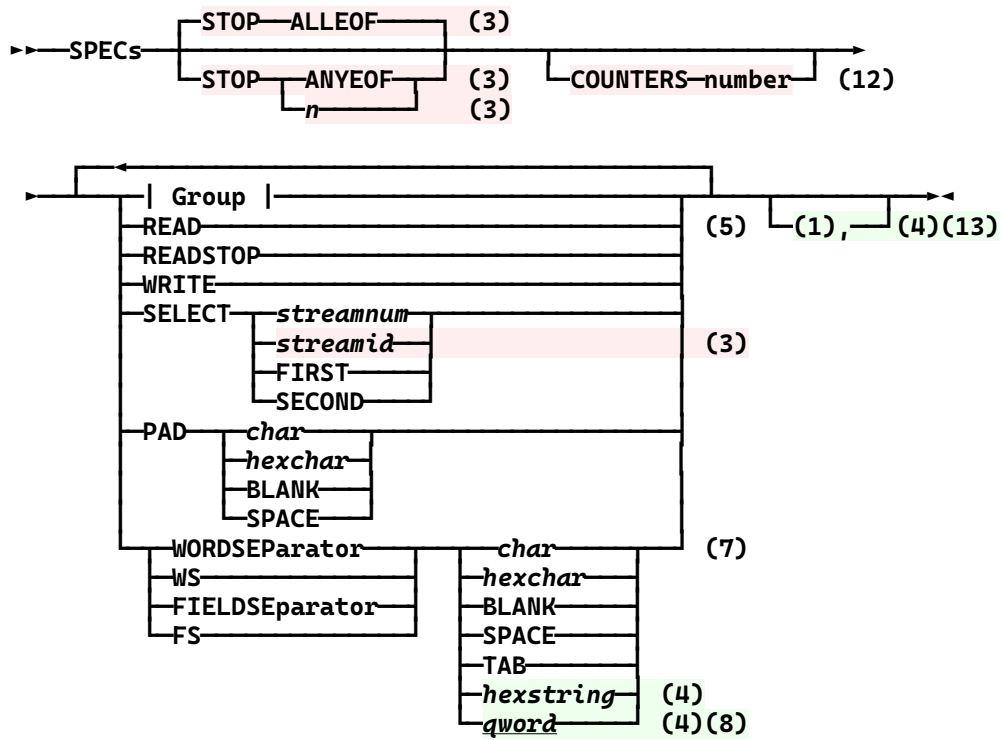
Space Words Like REXX



- (0) The order is the reverse of CHANGE!
- (1) the replacement char/string
- (2) the char/chars that will be stripped and replaced
- (3) NetRexx Pipelines only, not CMS. The dstring is treated as a single unit for stripping or replacing

spec
specs
4.05
Fields and
Separators
NetRexx and
COUNTERS
comma
separators

Rearrange Contents of Records



Group:

| Id | Input | Conversion | Output | Alignment | (1) |

Id:

| letter (2) (14) | (1) |

Input:

| Words (1) wnumberrange |

| Fields (1) fnumberrange |

| cnumberrange |

| delimitedString |

| Xhexstring |

| Hhexstring |

| Bbinstring |

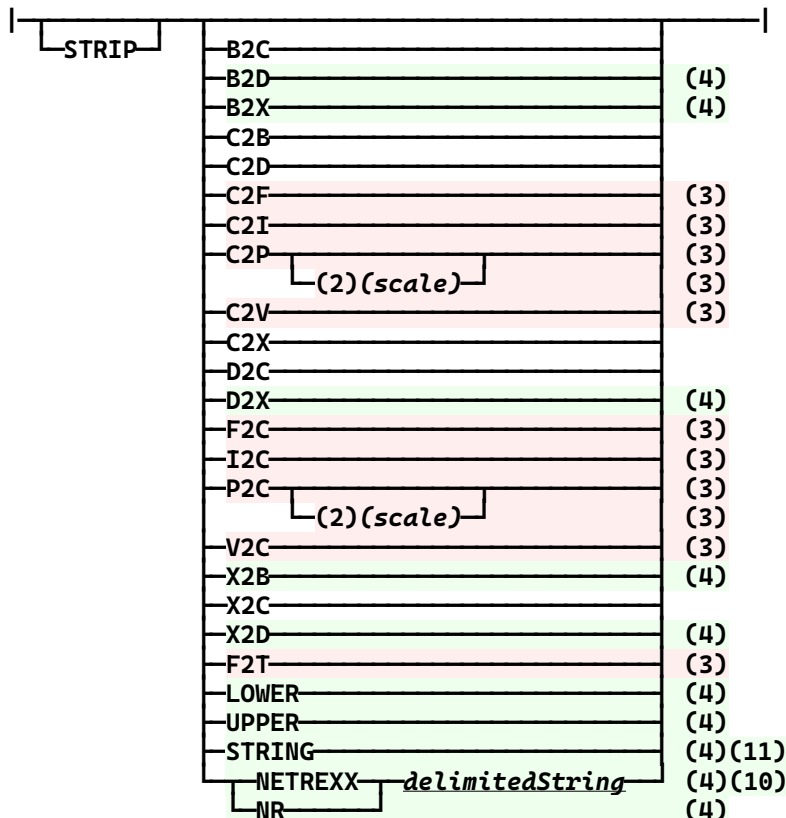
| RECNO | FROM 1 | BY 1 |

| NUMBER | FROM fromnum | BY bynum |

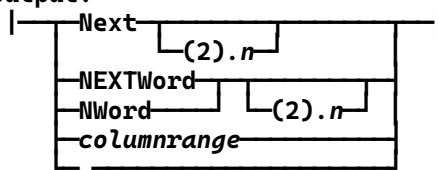
| TODclock | (3) |

| (4) (15) |

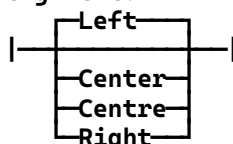
Conversion:



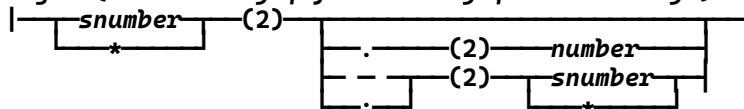
Output:



Alignment:



Ranges (cnumberrange, fnumberrange, wnumberrange):

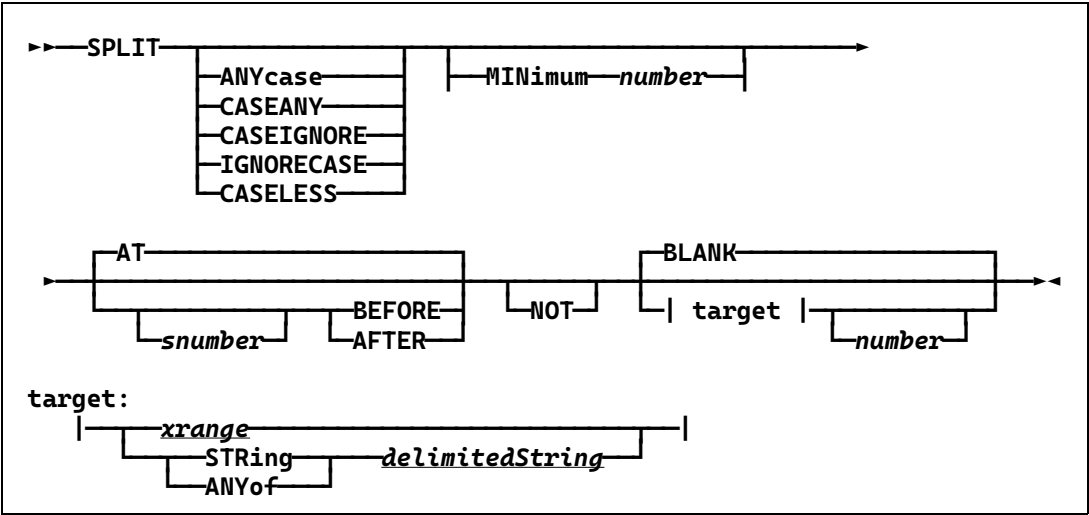


spill

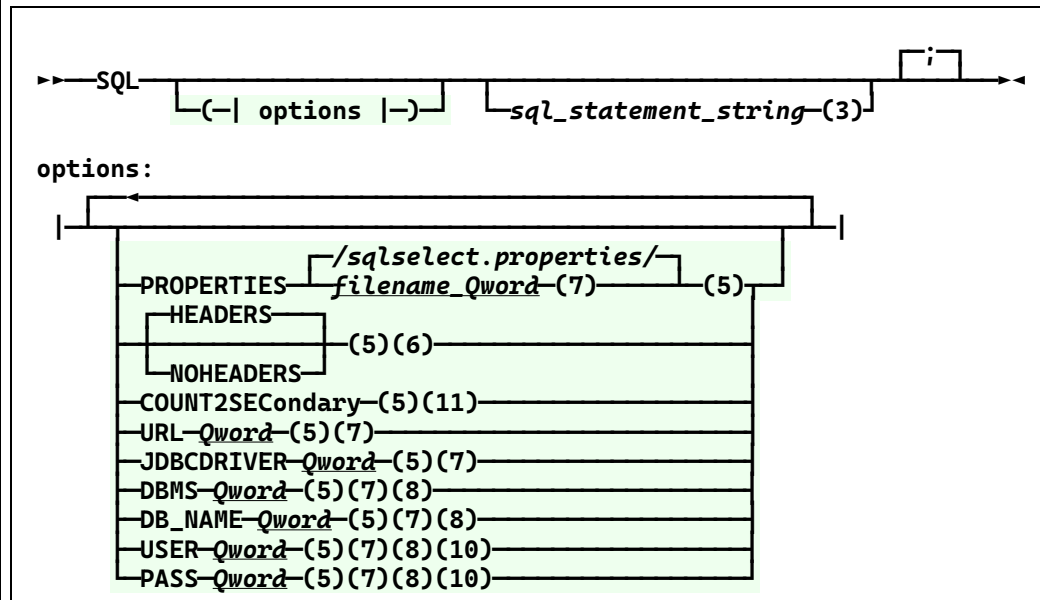
Spill Long Lines at Word Boundaries

- Not implemented in Netrex Pipelines.
- (1) Blanks are optional in this position.
- (2) Blanks are not allowed here.
- (3) CMS only. Not yet implemented in NetRexx Pipelines
- (4) NetRexx Pipelines only. Not yet implemented in CMS
- (5) NetRexx Pipelines only. READ is giving the same output as READSTOP when the streams are different length.
- (6) This senses if it is the first stage, but comment stages will fool it into not producing any output.
- (7) CMS Pipelines, without documenting it, places this right by default NetRexx Pipelines follows the documentation and places this left by default. Specify the alignment you want to override these defaults.
- (8) A *qword* is an optionally quoted word, with single or double marks. If it contains spaces or begins with a quote mark, it must be quoted. It can not start with a space (the quote mark will be considered a single character, and rest gibberish). If is unquoted and an even number of hexadecimal characters, it will be used as a hexchar or hexstring.
- (9) CMS has a mini-programming language built in. It uses Field Identifiers and Control Breaks, Counters, and Structured Data.
NetRexx does not yet have any of these features.
- (10) The delimited string is any valid NetRexx code. [Yes, you can get in trouble!] It is put into a method and executed for each record. The selected input data is in the variable DATA. The returned string is output. The

Split Records Relative to a Target



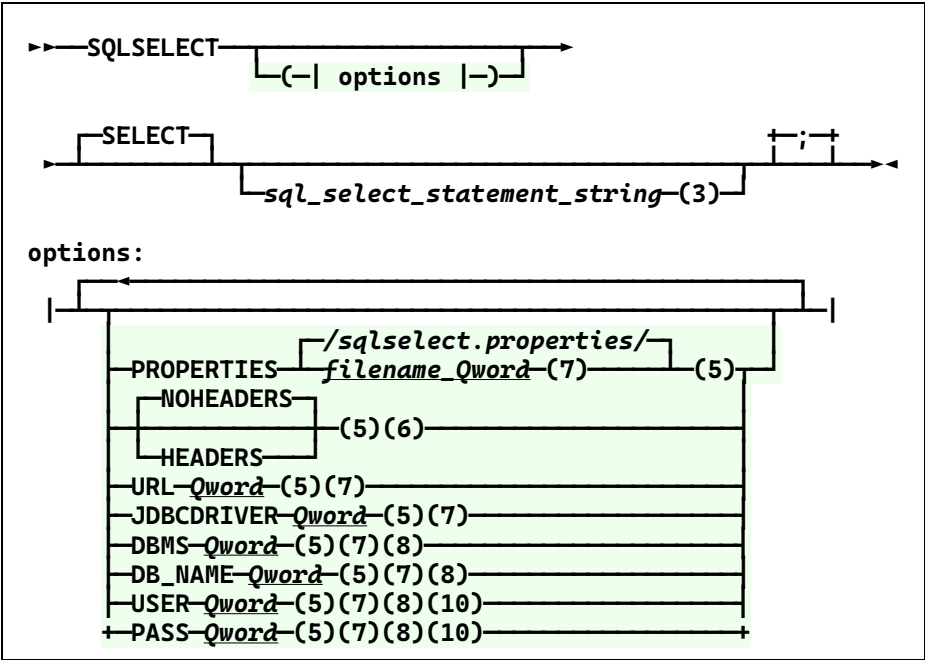
Interface to SQL



- uses jdbc to select from any jdbc enabled dbms
- properties file (sqlselect.properties default) is read from the secondary input stream to find jdbcdriver name, url, user, pass
- sample properties file:
- ```
#JDBC driver name
#Tue Feb 03 23:29:43 GMT+01:00 1998
jdbcdriver=com.imaginary.sql.msql.MsqlDriver
url=jdbc:msql://localhost:1114/TESTDB
the following are not needed for some DBMS, ex: SQLite
user=db_user_name
pass=password_for_db
```
- if this file is not found default (compiled in) values are used
- (1) when using a sql select \* (all columns) from the commandline, quote the query as in `java pipes.compiler (query) "sql select * from dept | console"`
- (2) the netrexx/jdbc combination is extremely case sensitive for column and table names
- (3) this sql\_select\_string executed, then statements are read from the primary input stream. this is optional in NetRexx Pipelines only.
- (4) CMS does not use the stream input
- (5) NetRexx Pipelines only
- (6) CMS Pipelines is implied HEADERS only.
- (7) A Qword is an optionally quoted word. If it contains spaces, it must be quoted.
- (8) EXPERIMENTAL Subject to change. DBMS is the kind of database, e.g. SQLite. DB\_name is the file name. These are used in place of URL and JDBC DRIVER. SQLite is the only one tested as of 8/15/20.
- (9) the SQLSELECT stage uses HEADERS as the default.
- (10) USER & PASS are needed for some DBMSs and not others, ex. SQLite.
- (11) the count or other output from non-select statements goes to the secondary output stream if connected, or is discarded. Otherwise it goes to the primary.
- 
- Priority order for URL, JDBC DRIVER and DBMS, DB\_NAME (first one found rules):
  1. option in the SQL command string
  2. from secondary input stream
  3. from "sql.properties" file or from file specified by PROPERTIES option
  4. Builtin

## Write the last 11 SQL Codes Received

- Not implemented in Netrexx Pipelines.



- (1) when using a `sqlselect *` (all columns) from the commandline, quote the query as in `java pipes.compiler` (query) `"sqlselect * from dept | console"`
- (2) the `netrexx/jdbc` combination is extremely case sensitive for column and table names
- (3) if no `sql_select_string` is specified, it is read from the primary input stream. this is optional in NetRexx Pipelines only. CMS does not use the stream input.
- (4) a maximum of only one record is ever read from the primary input stream.
- (5) NetRexx Pipelines only
- (6) CMS Pipelines is implied HEADERS only.
- (7) A Qword is an optionally quoted word. If it contains spaces, it must be quoted.
- (8) EXPERIMENTAL Subject to change. DBMS is the kind of database, e.g. SQLite. DB\_name is the file name. These are used in place of URL and JDBCDRIVER. SQLite is the only one tested as of 8/15/20.
- (9) the SQL stage uses NOHEADERS as the default.
- (10) USER & PASS are needed for some DBMSs and not others, ex. SQLite.
- Priority order for URL, JDBCDRIVER, DBMS, DB\_NAME, USER, & PASS (first one found rules):
  1. option in the SQL command string
  2. from secondary input stream
  3. from "sqlselect.properties" file or from file specified by PROPERTIES option
  4. Builtin

- Not implemented in Netrexx Pipelines.

- Not implemented in Netrexx Pipelines.

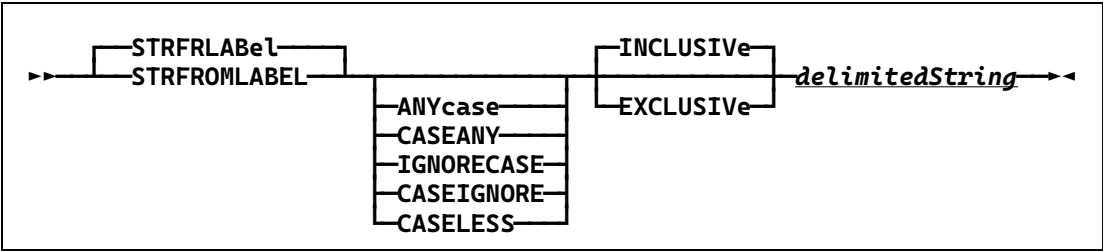
- Not implemented in Netrexx Pipelines.



|                                                                                 |                                                                                                                                                        |
|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>starsys</i>                                                                  | <b>Write Lines from a Two-way CP System Service</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>            |
| <i>state</i>                                                                    | <b>Provide Information about CMS Files</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                     |
| <i>state</i>                                                                    | <b>Verify that Data Set Exists</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                             |
| <i>statew</i>                                                                   | <b>Provide Information about Writable CMS Files</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>            |
| <i>stem</i>                                                                     | <b>Retrieve or Set Variables in a REXX or CLIST Variable Pool</b> <div> <p><b>NetRexx</b></p> <p><b>CMS</b></p> </div>                                 |
| <i>stfle</i>                                                                    | <b>Store Facilities List</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                   |
| <i>storage</i>                                                                  | <b>Read or Write Virtual Machine Storage</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                   |
| <i>strasmfind</i>                                                               | <b>Select Statements from an Assembler File as XEDIT Find</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>  |
| <i>strasmnfind</i>                                                              | <b>Select Statements from an Assembler File as XEDIT NFind</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul> |
| <i>strfind</i>                                                                  | <b>Select Lines by XEDIT Find Logic</b>                                                                                                                |
| <i>strfrlabel</i><br><i>strfrlabe</i><br><i>strfrlab</i><br><i>strfromlabel</i> | <b>Select Records from the First One with Leading String</b>                                                                                           |

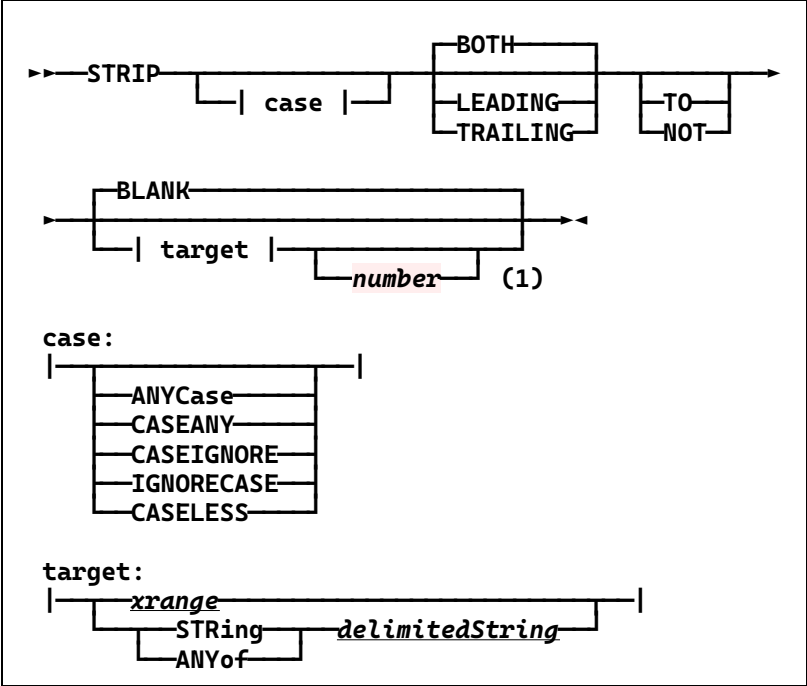
*strfromlabel*  
*strfrlabel*  
*strfrlabe*  
*strfrlab*

Select Records from the First One with Leading String



*strip*

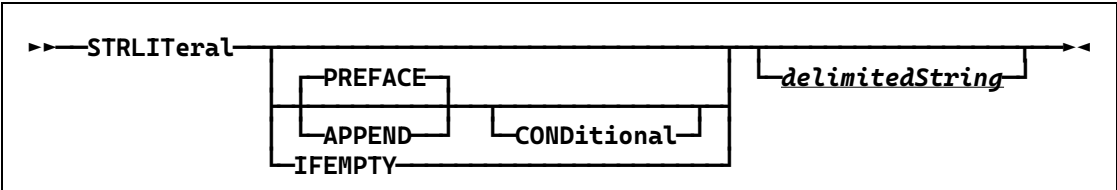
Remove Leading or Trailing Characters



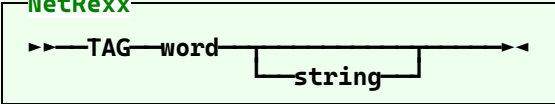
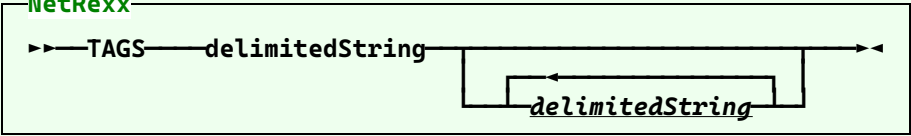
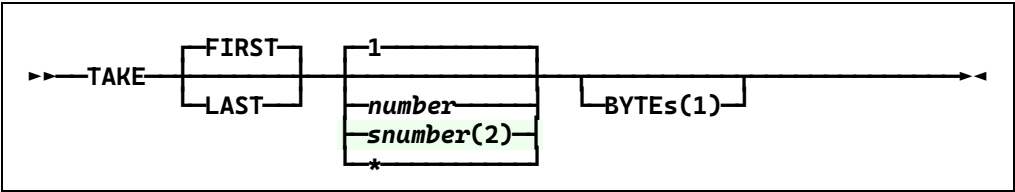
- (1) Not implemented in Netrexx Pipelines.

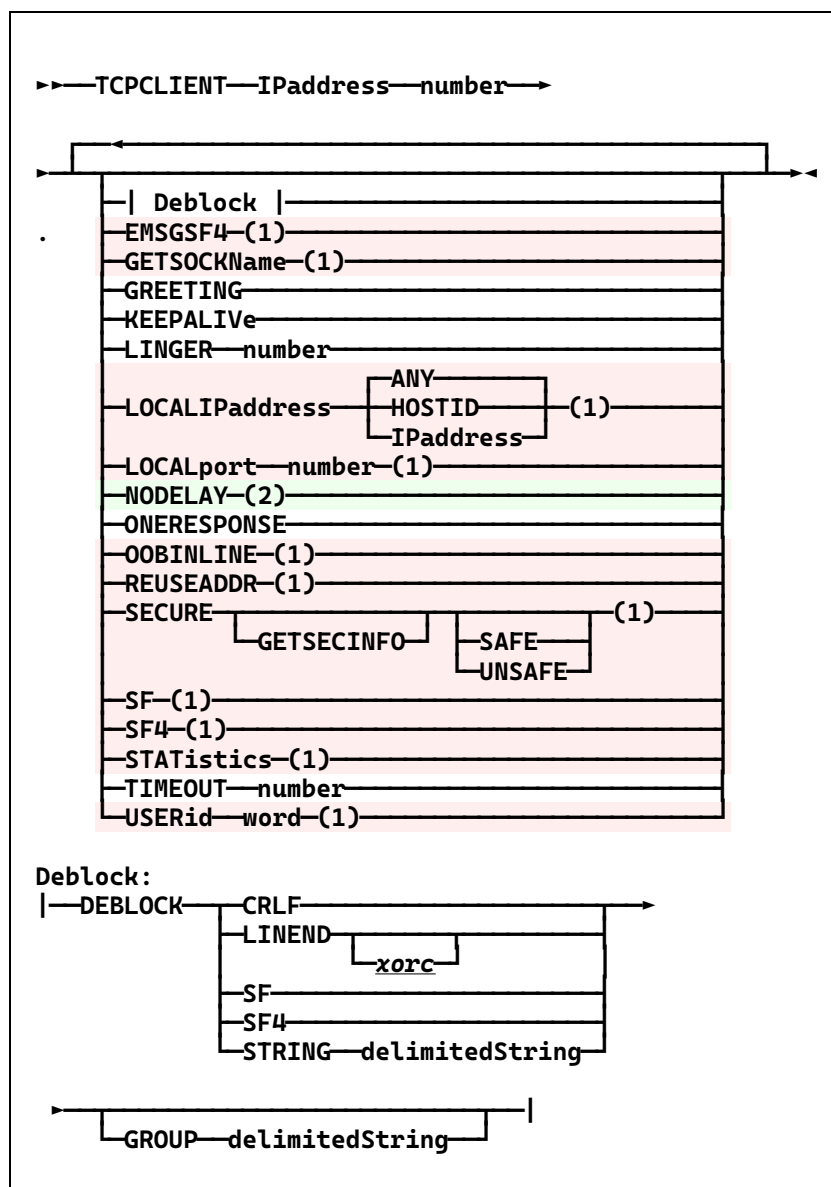
*strliteral*  
*strlitera*  
*strliter*  
*strlite*  
*strlit*

Write the Argument String



|                                                                                                                                       |                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <i>strnfind</i>                                                                                                                       | <b>Select Lines by XEDIT NFind Logic</b> <div> </div>                                                                                       |
| <i>strtolabel</i><br><i>strtolabe</i><br><i>strtolab</i>                                                                              | <b>Select Records to the First One with Leading String</b> <div> </div>                                                                     |
| <i>structure</i><br><i>struct</i>                                                                                                     | <b>Manage Structure Definitions</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>               |
| <i>strwhilelable</i><br><i>strwhilelabl</i><br><i>strwhilelab</i><br><i>strwhilela</i><br><i>strwhilel</i><br><i>strwhile</i><br>3.09 | <b>Select Run of Records with Leading String</b> <div> </div>                                                                               |
| <i>stsi</i>                                                                                                                           | <b>Store System Information</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>                   |
| <i>subcom</i>                                                                                                                         | <b>Issue Commands to a Subcommand Environment</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul> |
| <i>substring</i><br><i>substr</i>                                                                                                     | <b>Write substring of record</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>                  |
| <i>synchronise</i><br><i>sync</i><br><i>synchronize</i>                                                                               | <b>Synchronise Records on Multiple Streams</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>    |
| <i>synchronize</i><br><i>sync</i><br><i>synchronise</i>                                                                               | <b>Synchronise Records on Multiple Streams</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>    |
| <i>sysdsn</i>                                                                                                                         | <b>Test whether Data Set Exists</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>               |
| <i>sysout</i>                                                                                                                         | <b>Write System Output Data Set</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>               |
| <i>sysvar</i>                                                                                                                         | <b>Write System Variables to the Pipeline</b> <ul style="list-style-type: none"> <li>• Not implemented in Netrexx Pipelines.</li> </ul>     |

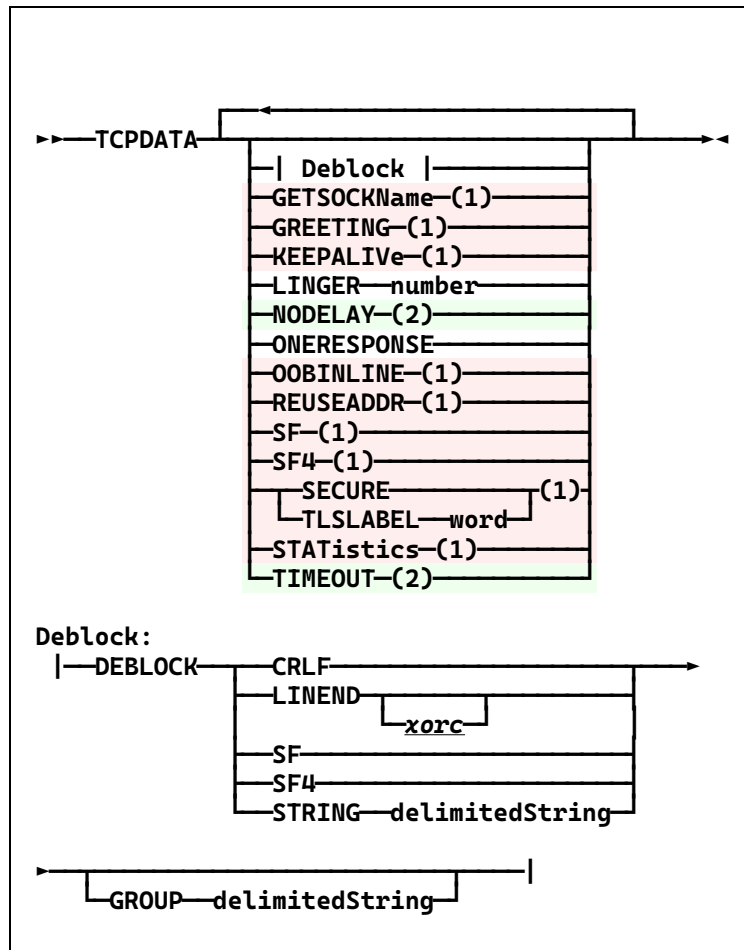
|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>tag</i><br>3.11  | <p><b>Surrounds Input Records with a HTML tag and its End Tag</b></p> <p><b>NetRexx</b></p>  <ul style="list-style-type: none"> <li>Outputs a record: &lt;word string&gt;, then passes through all records on its primary input, and finally a record: &lt;/word&gt;.</li> </ul>                                                                                                                                        |
| <i>tags</i><br>3.11 | <p><b>Surrounds Input Records with HTML tags and their End Tags</b></p> <p><b>NetRexx</b></p>  <ul style="list-style-type: none"> <li>Outputs a record for each delimitedString: &lt;delimitedString&gt;, then passes through all records on its primary input, and finally a record for each, in reverse order: &lt;/first_word_of_delimitedString&gt;.</li> <li>Any delimitedString may be a single word.</li> </ul> |
| <i>take</i>         | <p><b>Select Records from the Beginning or End of the File</b></p>  <ul style="list-style-type: none"> <li>(1) CMS must be BYTES</li> <li>(2) Not CMS; NetRexx Pipelines: minus reverses first/last</li> </ul>                                                                                                                                                                                                         |
| <i>tape</i>         | <p><b>Read or Write Tapes</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                |
| <i>tcpchsum</i>     | <p><b>Compute One's complement Checksum of a Message</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                     |



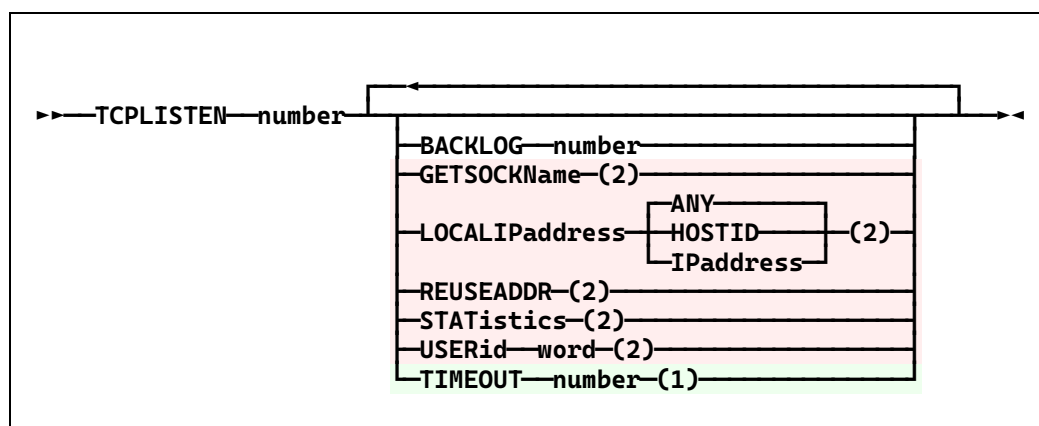
- (1) CMS Pipelines Only.
- (2) NetRexx Pipelines Only.

The options implemented are similar to the CMS definition.

- **linger** - wait a bit before terminating the last read (units SECONDS)
- **timeout** - wait this long before timing reads out (units MS)
- **deblock** - If deblock is omitted a copy stage is used.
- **group** - similar to CMS. A delimited string containing a stage is expected. You can use a run of stages, but its is dangerous since you don't know the stage sep character being used...
- **greeting** - expect a greeting message and discard it
- **nodelay** - use the nodelay option
- **keepalive** - enable keep alive socket option
- **oneresponse** - synchronize cmds/replys



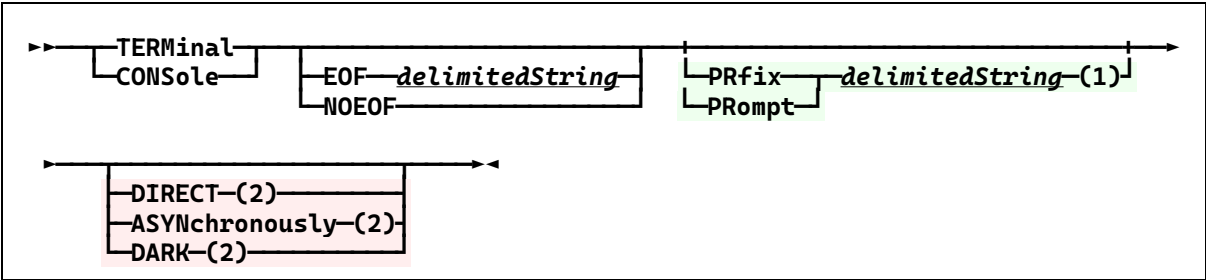
- Simple tcpdata implementation.
- (1) CMS Pipelines Only
- (2) NetRexx Pipelines Only
  - linger - wait a bit before terminating the last read (units SECONDS)
  - timeout - wait this long before timing reads out (units MS)
  - deblock - If deblock is omitted a copy stage is used.
  - group - similar to cms. A delimited string containing a stage is expected. You can use a run of stages, but its is dangerous since you to know the stage sep character being used...
  - nodelay - use the nodelay option
  - onerresponse - synchronize requests/replies



- (1) NetRexx Pipelines only.
- (2) CMS Pipelines only.
- Simple tcplisten implementation. You can only supply the port and a timeout value. Its ignored unless tcplisten's output stream has been severed, in which case tcplisten terminates.
- If input stream 0 is connected, tcplisten does a peekto before calling the accept method. The object is consumed after the output of the socket object returns.

terminal  
termina  
termina  
termin  
termi  
term  
console  
consol  
conso  
cons  
cons  
3.11

### Read or Write the Terminal in Line Mode



- (1) NetRexx only  
On first stage, delimitedString is put out as a prompt  
On other stages, each line is prefixed with delimitedString  
Outout to next stage does NOT include delimitedString  
Either keyword can be used for either stage
- (2) CMS only

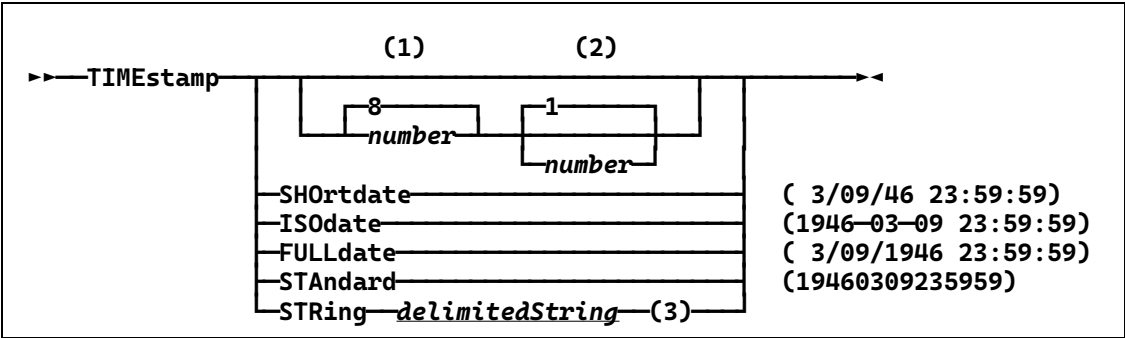
threeway

### Split record three ways

- Not implemented in NetrexX Pipelines.

timestamp  
timestam  
timesta  
timest  
times  
time

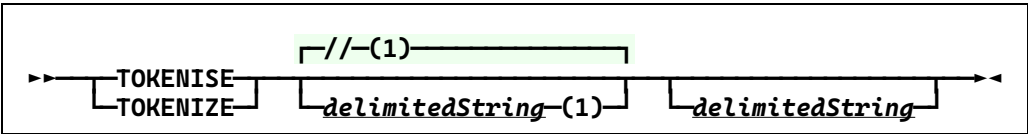
### Prefix the Date and Time to Records



- (1) First character, from right, to include, <= 16
- (2) Count of characters to include. <= 16 - (1). Default = (1)
- (3)
  - %% A single %.
  - %Y Four digits year including century (0000-9999).
  - %y Two-digit year of century (00-99).
  - %m Two-digit month (01-12).
  - %n Two-digit month with initial zero changed to blank ( 1-12).
  - %d Two-digit day of month (01-31).
  - %e Two-digit day of month with initial zero changed to blank ( 1-31).
  - %j Julian day of year (001-366).
  - %H Hour, 24-hour clock (00-23).
  - %k Hour, 24-hour clock with leading zero blank ( 0-23).
  - %M Minute (00-59).
  - %S Second (00-60).
  - %F Equivalent to %Y-%m-%d (the ISO 8601 date format).
  - %T Short for %H:%M:%S.
  - %t Tens and hundredth of a second (00-99).

tokenise  
tokenize

### Tokenise Records

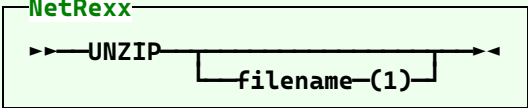
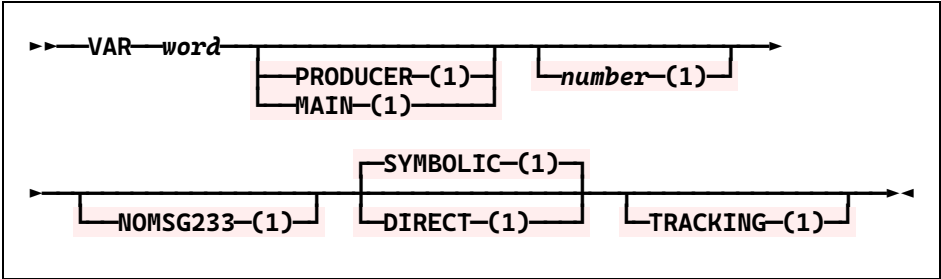


- (1) In CMS Pipelines, the first delimited string is required. In NetRexX Pipelines, it defaults to // if no second string.

|                            |                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------|
| tolabel<br>tolabe<br>tolab | <div> Select Records to the First One with Leading String <div> <div> ▶▶TOLABel </div> <div> string </div> </div> </div> |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------|



|                                                                                   |                                                                                                                                               |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <i>trfread</i>                                                                    | <b>Read a Trace File</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                              |
| <i>truncate</i><br><i>truncat</i><br><i>trunca</i><br><i>trunc</i><br><i>chop</i> | <b>Truncate the Record</b>                                                                                                                    |
| <i>tso</i>                                                                        | <b>Issue TSO Commands, Write Response to Pipeline</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul> |
| <i>udp</i>                                                                        | <b>Read and Write an UDP Port</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                     |
| <i>unique</i><br><i>uniqu</i><br><i>uniq</i>                                      | <b>Discard or Retain Duplicate Lines</b>                                                                                                      |

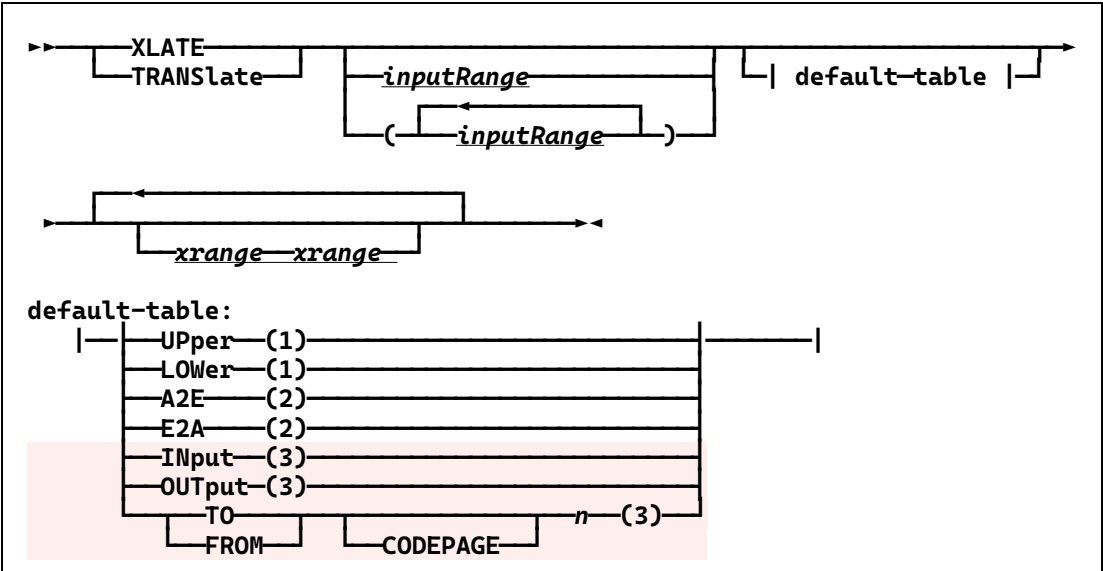
|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>unpack</i>     | <b>Unpack a Packed File</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>untab</i>      | <b>Replace Tabulate Characters with Blanks</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>unzip</i>      | <b>Extract Files From a ZIP Archive</b> <div>  <pre> graph LR     In(( )) --&gt; UNZIP     UNZIP --&gt; Out(( ))     UNZIP --&gt; F1[filename-(1)]           </pre> </div> <p>Notes:</p> <ul style="list-style-type: none"> <li>(1) If filename is not specified, it is read from the primary input stream. Succeeding input objects are ignored.</li> <li>The extracted file names are passed to the primary output stream.</li> <li>Any existing files will be replaced.</li> <li>This is NetRexx Pipelines only.</li> </ul>                                                                                                 |
| <i>update</i>     | <b>Apply an Update File</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>urldeblock</i> | <b>Process Universal Resource Locator</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>uro</i>        | <b>Write Unit Record Output</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>utf</i>        | <b>Convert between UTF-8, UTF-16, and UTF-32</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>var</i>        | <b>Retrieve or Set a Variable in a REXX or CLIST Variable Pool</b> <div>  <pre> graph LR     In(( )) --&gt; VAR_word[VAR word]     VAR_word --&gt; Out(( ))     VAR_word --&gt; P1[PRODUCER-(1)]     VAR_word --&gt; M1[MAIN-(1)]     VAR_word --&gt; N1[number-(1)]     Out --&gt; S1[SYMBOLIC-(1)]     Out --&gt; N2[NOMSG233-(1)]     Out --&gt; D1[DIRECT-(1)]     Out --&gt; T1[TRACKING-(1)]     Out --&gt; Out2(( ))           </pre> </div> <ul style="list-style-type: none"> <li>In NetRexx Pipelines, this can only read vars. It must be the first stage in a pipe.</li> <li>(1) CMS Pipelines only.</li> </ul> |
| <i>vardrop</i>    | <b>Drop Variables in a REXX Variable Pool</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>varfetch</i>   | <b>Fetch Variables in a REXX or CLIST Variable Pool</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>varload</i>    | <b>Set Variables in a REXX or CLIST Variable Pool</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>varover</i><br>3.09 | <p><b>Write the Values of Stems</b></p> <div> <p>NetRexx</p> <pre>►—VAROVER—varName—◄</pre> </div> <ul style="list-style-type: none"> <li>NetRexx Pipelines only; not CMS Pipelines</li> <li>If the secondary output stream is connected, the root is passed on it.</li> </ul>                                                                                                                                                                                                                                                                                                                                     |
| <i>varset</i>          | <p><b>Set Variables in a REXX or CLIST Variable Pool</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>vchar</i>           | <p><b>Recode Characters to Different Length</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>vector</i>          | <p><b>Read or Write an Array of Vectors</b></p> <div> <p>NetRexx</p> <pre>►—VECTOR—name—◄</pre> </div> <ul style="list-style-type: none"> <li>Pipes for NetRexx only.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>vectora</i>         | <p><b>Add to an Array of Vectors</b></p> <div> <p>NetRexx</p> <pre>►—VECTORA—name—◄</pre> </div> <ul style="list-style-type: none"> <li>Pipes for NetRexx only.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>vectorr</i>         | <p><b>Read From an Array of Vectors</b></p> <div> <p>NetRexx</p> <pre>►—VECTORR—name—◄</pre> </div> <ul style="list-style-type: none"> <li>Pipes for NetRexx only.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>vectorw</i>         | <p><b>Write to an Array of Vectors</b></p> <div> <p>NetRexx</p> <pre>►—VECTORW—name—◄</pre> </div> <ul style="list-style-type: none"> <li>Pipes for NetRexx only.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <i>verify</i><br>3.09  | <p><b>Verify that Record Contains only Specified Characters</b></p> <div> <pre> ►—VERIFY— ├── ANYCASE ─┤ ├── CASEANY ─┤ ├── CASEIGNORE ─┤ ├── IGNORECASE ─┤ └── CASELESS ─┘ ├── inputRange ─┤ └── delimitedString ─┤ (1)                     character-range (1) </pre> </div> <ul style="list-style-type: none"> <li>(1) NetRexx Pipelines only</li> <li>(1) <i>character-range</i> is <i>xorc-xorc</i></li> <li>(1) Examples: A-Z 0-9 c-g a4-ba; 16-bit Unicode characters or hex numbers</li> <li>(1) Any number greater than zero, any order, of delimitedStrings and character-ranges are allowed.</li> </ul> |
| <i>vmc</i>             | <p><b>Write VMCF Reply</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i>vmcdata</i>         | <p><b>Receive, Reply, or Reject a Send or Send/receive Request</b></p> <ul style="list-style-type: none"> <li>Not implemented in NetrexX Pipelines.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                           |                                                                                                                                          |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <i>vmclient</i>           | <b>Send VMCF Requests</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                        |
| <i>vmclisten</i>          | <b>Listen for VMCF Requests</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                  |
| <i>waitdev</i>            | <b>Wait for an Interrupt from a Device</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>       |
| <i>warp</i>               | <b>Pipeline Wormhole</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                         |
| <i>warplist</i>           | <b>List Wormholes</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>                            |
| <i>whilelabel</i><br>3.09 | <b>Select Run of Records with Leading String</b> <div> </div>                                                                            |
| <i>wildcard</i>           | <b>Select Records Matching a Pattern</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>         |
| <i>writepds</i>           | <b>Store Members into a Partitioned Data Set</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul> |
| <i>xab</i>                | <b>Read or Write External Attribute Buffers</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>  |
| <i>xedit</i>              | <b>Read or Write a File in the XEDIT Ring</b> <ul style="list-style-type: none"> <li>Not implemented in Netrexx Pipelines.</li> </ul>    |

xl  
late  
trans  
late  
trans  
lat  
trans  
la  
trans  
l  
trans

Transliterate Contents of Records



Notes:

- (1) UTF-16 (ASCII) in NJPipes, probably EBCDIC in CMS.
- (2) In Netrex Pipelines. EBCDIC to ASCII or ASCII to EBCDIC. Maybe in CMS, the documentation is unclear.
- (3) Not yet in NetRex Pipelines
- [4] NetRex Pipelines only: The secondary input stream is not yet supported.

xmsg

Issue XEDIT Messages

- Not implemented in Netrex Pipelines.

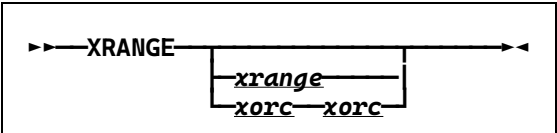
xpdhi

Expand Highlighting to Space between Words

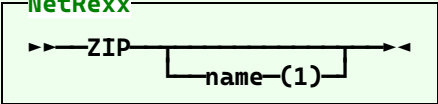
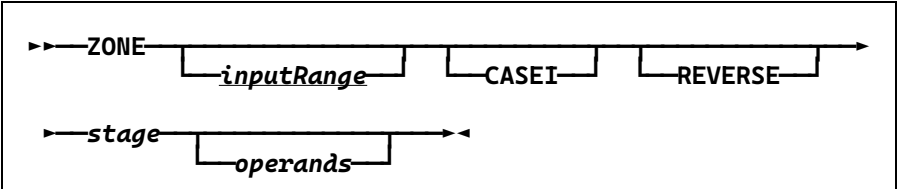
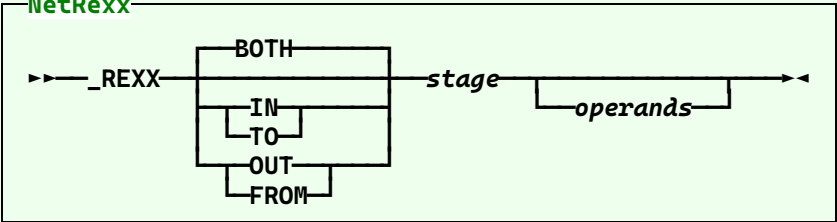
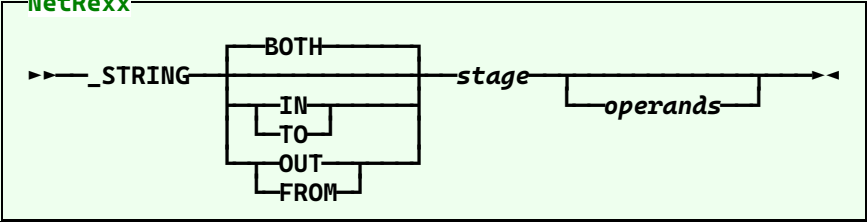
- Not implemented in Netrex Pipelines.

xrange  
3.09

Write a Range of Characters



- NetRex uses UTF-16 (ASCII) and CMS uses EBCDIC

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>zip</i>     | <p><b>Add Files To a new ZIP Archive</b></p> <p><b>NetRexx</b></p>  <ul style="list-style-type: none"> <li>• (1) name is the zip file name. If not provided, the first entry on the primary input stream is used.</li> <li>• (1) If no extension is provided in name, ".ZIP" is added.</li> <li>• (1) Any existing file is replaced.</li> <li>• Subsequent records on the primary input stream are filenames to be added to the Zip archive file.</li> <li>• File names added passed out on primary out stream.</li> <li>• NetRexx Pipelines only.</li> </ul> |
| <i>zone</i>    | <p><b>Run Selection Stage on Subset of Input Record</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>_rexx</i>   | <p><b>Cast Input and/or Output of a Stage to Type Rexx</b></p> <p><b>NetRexx</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>_string</i> | <p><b>Cast Input and/or Output of a Stage to Type String</b></p> <p><b>NetRexx</b></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |