```
--- kexx.kld    2020-04-11 13:13:32.356654000 +0200
+++ rexx.kld    2020-04-14 00:12:09.996717400 +0200
@@ -1,5 +1,5 @@
-* KEXX.KLD - KEDIT Language Definition for KEXX macros and KML
-* files (Kedit Macro Library), mostly identical to the KeditW32
+* REXX.KLD - KEDIT Language Definition for classic (ANSI) REXX
+* and classic ooRexx features, mostly identical to the KeditW32
  * built-in REXX.KLD (Version 1.1 - May 2007).

  :case
@@ -9,35 +9,34 @@
  :number
   rexx
  :identifier
-* skip KEXX legacy non-standard @#$
-*[a-zA-Z!?_@#$]    [a-zA-Z0-9.!?_@#$]
- [a-zA-Z!?_]       [a-zA-Z0-9.!?_]
+* experimental . at begin for .RS etc.
+ [a-zA-Z.!?_]      [a-zA-Z0-9.!?_]
  :string
   single
   double
  :comment
   paired     /* */ nest
- line       *     firstnonblank
-* The KML "header" is a colon followed by the macro name, e.g.,
-* :: defines macro : for key :.  A ::* would define a useless
-* macro :*, this is used for KML comments.  Unlike * and /* */
-* comments ::* comments are not stored in the Kedit macrospace.
- line       ::*   column 1
+* for ooRexx:
+ line       --    any
  :header
- line       :     column 1
+* for ooRexx directives:
+ line      ::     firstnonblank
+* (in)famous line 1 hacks:
+ line ï»¿         column 1
+ line #!          column 1
  :label
-* KEXX labels use column 1:
- delimiter :       column 1
-*delimiter :       any
+ delimiter :       any
  :match
- (          )
- do,select  end   when,otherwise
+ (                 )
+* loop and [] added for ooRexx:
+ [                 ]
+ do,loop,select    end   case,when,otherwise
  :keyword
-* No REXX address, push, queue
-*address     type 5
+ address     type 5
   arg         type 7
   by          type 2
```

```
  call         type 5
+ caseless     type 2
  digits       type 2
  do           type 5
  drop         type 1
@@ -45,7 +44,9 @@
  end          type 1
  engineering  type 2
  exit         type 1
- expose       type 2
+* for ooRexx:
+*expose       type 2
+ expose       type 3
  for          type 2
  forever      type 2
  form         type 2
@@ -53,6 +54,7 @@
  if           type 1
  interpret    type 1
  iterate      type 1
+ label        type 2
  leave        type 1
  linein       type 2
  name         type 2
@@ -65,9 +67,8 @@
  parse        type 5
  procedure    type 5
  pull         type 7
-* No REXX address, push, queue
-*push         type 1
-*queue        type 1
+ push         type 1
+ queue        type 1
  return       type 1
  say          type 1
  scientific   type 2
@@ -82,23 +83,86 @@
  value        type 2
  var          type 2
  version      type 2
- when         type 1
+* for ooRexx guard:
+*when         type 1
+ when         type 5
  while        type 2
  with         type 2
-* KEXX conditions:
+* REXX conditions:
  error        type 2
  failure      type 2
  halt         type 2
-*lostdigits   type 2
-*notready     type 2
+ lostdigits   type 2
+ notready     type 2
  novalue      type 2
  syntax       type 2
```

```
+* disabled sub-keywords of 'address'
+*append      type 2
+*input       type 2
+*normal      type 2
+*output      type 2
+*path        type 2
+*replace     type 2
+*stem        type 2
+*stream      type 2
+*system      type 2
+* ooRexx keywords:
+ additional  type 2
+ any         type 2
+ arguments   type 2
+ array       type 2
+ case        type 2
+ class       type 2
+ continue    type 2
+ counter     type 2
+ description type 2
+ forward     type 5
+ guard       type 5
+ index       type 2
+ item        type 2
+ local       type 2
+ loop        type 5
+ message     type 2
+ nomethod    type 2
+ nostring    type 2
+ over        type 2
+ propagate   type 2
+ raise       type 5
+ reply       type 1
+ strict      type 2
+ use         type 5
+ user        type 2
+* ANSI REXX special:
+ rc          type 1
+ result      type 1
+ sigl        type 1
+ .MN         type 1
+ .RC         type 1
+ .RESULT     type 1
+ .RS         type 1
+* Common special:
+ .ENDOFLINE  type 1
+ .LINE       type 1
+* Regina special:
+ .DIRSEP     type 1
+ .FILE       type 1
+* ooRexx special:
+ self        type 1
+ super       type 1
+ .FALSE      type 1
+ .NIL        type 1
+ .TRUE       type 1
 :postcompare
```

```
 *See 'ecolor': 9 blue (like keywords), 8 black, 6 dark cyan (strings), 1 red.
 *The sort order is significant, anything already handled above does not work,
 *e.g., most strings, identifiers, keywords, and comments are already colored.
-*red for backtick + deprecated
- class [`~^@#$]   alternate 1
-*red controls and non-ASCII
+
+* logical NOT U+00AC is valid (assuming Latin-1 or similar here, not UTF-8):
+ text ¬           alternate 8
+* red for comma, tricky at the end of a line
+ text ,           alternate 1
+* red for backtick + obsolete:
+ class [`^@#$]    alternate 1
+* red controls and non-ASCII (outside of strings and comments)
  class [~!--]     alternate 1
-*KEXX rejects U+00AC logical NOT
+
```