

```
FA_Beta DXO PLS\Hierarchical keyword handling\Hierarchical keyword handling\Pseudo code V03-1.txt
07/07/2022 09:15:01 5,241 bytes Everything Else - ANSI - PC
Access 'Items'
** Locate any keywords for an 'Items' entry via the 'ItemsKeyword' structure
** A keyword can fall into one of the following categories:-
** A keyword found from 'ItemsKeywords', i.e. an assigned keyword
** which may be:-
** A simple keyword, designated by a 'ParentId' of NULL
** An hierarchical keyword, the "leaf" of an hierarchical keyword (which may be an element of a larger hierarchical keyword)
** and then via the 'Keywords' Table accessed using the 'KeywordId' from 'ItemsKeywords'
** An hierarchical keyword found by a 'ParentId' pointer from another keyword and containing a 'ParentId'
** An hierarchical keyword found by a 'ParentId' pointer from another keyword and containing a 'ParentId' = NULL - i.e. the head
...
ACCESS 'ItemsKeywords' VIA the 'Itemid' value from 'Items'
** Start reconstructing any hierarchical keywords and populating the 'dc' and 'hr' fields appropriately
...
PROCESS EACH ASSIGNED KEYWORD
FOR EACH entry in 'ItemsKeyword' associated with the 'Itemid' THEN
** Taking each ASSIGNED keyword in turn (currently EVERY item in 'ItemsKeywords')
...
Locate the entry in the 'Keywords' Table
ACCESS 'Keywords' using the 'KeywordId' value from 'ItemsKeywords' (NOTFOUND = 'Unexpected ERROR')
** Determine keyword type by it's context, i.e. by the 'ParentId' setting etc.
IF 'ParentId' = 'NULL' THEN
PROCESS A SIMPLE KEYWORD
** 'NULL' is used for simple keywords, i.e. it is not part of an hierarchy or it is actually the head of a hierarchy
but the hierarchy traversing code is included separately below
** Typically every simple key is included in the 'dc' field but if the process is fully parameter driven
it becomes possible to configure a scenario (Template) where they can be omitted!
...
ADD 'Value' from 'Keywords' TO 'dc Keywords'
** Consider whether it is correct to include that keyword in the 'hr' fields as DxPL and others do currently?
(Unconditional):
ADD 'Value' from 'Keywords' TO 'hr Keywords' <-----@Joanna
(end Unconditional)
ELSE
** 'ParentId' Not NULL so part of an hierarchical keyword (in this case the "leaf" keyword)
PROCESS A LEAF HIERARCHICAL KEYWORD ELEMENT
** Start a new hierarchical keyword reconstruction from last to first, e.g. D
SET 'Hierarchical Keyword' = 'Value' from 'Keywords'
ADD | TO front of 'Hierarchical keyword'
...
This is the "leaf" keyword in the hierarchy, e.g. D in A|B|C|D (A>B>C>D) or an assigned sub-element hierarchy
The leaf will arguably ALWAYS be considered for the 'dc' keywords even for DxO PLS.2.0 option
(Unconditional):
ADD 'Value' from 'Keywords' TO 'dc Keywords'
(end Unconditional)
** Optionally add 'Value' from 'Keywords' to 'hr' Keywords!? <-----@Joanna
(Conditional):
ADD 'Value' from 'Keywords' TO 'hr Keywords'
(End-Conditional)
...
Using the 'ParentId' access the linked keyword in the 'Keywords' Table
FOR each entry in 'Keywords' Table found using the 'ParentId' from a 'Keywords' Table entry
ACCESS 'Keywords' VIA 'ParentId' of 'Keywords' Table entry
IF 'ParentId' = NULL Then
** This is the head keyword of the hierarchical keyword
PROCESS HEAD OF AN HIERARCHICAL KEYWORD
** Add keyword to the front of a reconstruction of the hierarchical keyword from leaf to top, i.e. in front of the previous k
ADD 'Value' from 'Keywords' TO front of 'Hierarchical keyword'
** Reconstruction of hierarchical keyword is now complete and the head node should be added to the 'hr' keywords <-----@Joanna
(Unconditional):
ADD 'Hierarchical keyword' TO 'hr Keywords'
...
Optionally add the keyword to the 'dc' keys this is different between pre PLS.2.0 & post PLS.2.0
(Conditional):
ADD 'Value' from 'Keywords' TO 'dc Keywords'
(End Conditional)
** Add keyword to 'hr Keywords', typically done with the head of an hierarchical keyword <-----@Joanna
(Unconditional):
ADD 'Value' from 'Keywords' TO 'hr Keywords'
(end Unconditional)
ELSE
** This is an intermediate hierarchical keyword
PROCESS INTERMEDIATE HIERARCHICAL KEYWORD
ADD 'Value' from 'Keywords' TO front of 'Hierarchical keyword'
ADD | TO front of 'Hierarchical keyword'
...
Optionally add the keyword to the 'dc' keys - this is different between pre PLS.2.0 & post PLS.2.0
FA_Beta DXO PLS\Hierarchical keyword handling\Hierarchical keyword handling\Pseudo code V03-3.txt
24/07/2022 12:38:02 9,074 bytes Everything Else - ANSI - PC
Access 'Items'
** Locate any keywords for an 'Items' entry via the 'ItemsKeyword' structure
** A keyword can fall into one of the following categories:-
** A keyword found from 'ItemsKeywords', i.e. an assigned keyword
** which may be:-
** A simple keyword, designated by a 'ParentId' of NULL
** An hierarchical keyword, the "leaf" of an hierarchical keyword (which may be an element of a larger hierarchical keyword)
** and then via the 'Keywords' Table accessed using the 'KeywordId' from 'ItemsKeywords'
** An hierarchical keyword found by a 'ParentId' pointer from another keyword and containing a 'ParentId'
** An hierarchical keyword found by a 'ParentId' pointer from another keyword and containing a 'ParentId' = NULL - i.e. the head
...
** Load the keyword format template (KFT) from the table using, the identifier or the template # or ?
18 READ 'KeywordsTemplate' from idx_KeywordTemplate AT 'KeywordTemplateId' =
19 **
20 ** = 'user Default KFT'
21 ** = 'DxPL Default KFT'
22 ** = 'Metadata Write KFT'
23 ** = 'Export Write KFT'
24 ** = etc. as appropriate
...
ACCESS 'ItemsKeywords' VIA the 'Itemid' value from 'Items'
** Start reconstructing any hierarchical keywords and populating the 'dc' and 'hr' fields appropriately
...
PROCESS EACH ASSIGNED KEYWORD
FOR EACH entry in 'ItemsKeyword' associated with the 'Itemid' THEN
** Taking each ASSIGNED keyword in turn (currently EVERY item in 'ItemsKeywords')
** But for Win 10 this will only be one entry for the last or Leaf keyword component
...
Locate the entry in the 'Keywords' Table
ACCESS 'Keywords' using the 'KeywordId' value from 'ItemsKeywords' (NOTFOUND = 'Unexpected ERROR')
** Determine keyword type by it's context, i.e. by the 'ParentId' setting etc.
IF 'ParentId' = 'NULL' THEN
PROCESS A SIMPLE KEYWORD
** 'NULL' is used for simple keywords, i.e. it is not part of an hierarchy or it is actually the head of a hierarchy
but the hierarchy traversing code is included separately below
** Typically every simple key is included in the 'dc' field but if the process is fully parameter driven
it becomes possible to configure a scenario (Template) where they can be omitted!
...
(Conditional on Parameter #1 - essentially unconditional but...)
IF 'Parameter #1' = "A" THEN
ADD ['Value' from 'Keywords'] TO 'dc Keywords'
END_IF
(End-Conditional)
...
** Consider whether it is correct to include that keyword in the 'hr' fields as DxPL and others do currently?
(Conditional on Parameter #2):
IF 'Parameter #2' = "A" THEN
ADD ['Value' from 'Keywords'] TO 'hr Keywords'
END_IF
(End Conditional)
ELSE
** 'ParentId' Not NULL so part of an hierarchical keyword (in this case the "leaf" keyword)
PROCESS A LEAF HIERARCHICAL KEYWORD ELEMENT
** Start a new hierarchical keyword reconstruction from last to first, e.g. D
SET 'Hierarchical Keyword' = ['Value' from 'Keywords']
** Processing an hierarchical component and an hierarchical keyword can be made generic so the values can between
Space/empty = do nothing
** "A" = include ALL
** "F" = include First, i.e. "A" from "A/B/C/D"
** "L" = include Last (Leaf), i.e. "D" from "A|B|C|D"
...
This is the "leaf" keyword in the hierarchy, e.g. D in A|B|C|D (A>B>C>D) or an assigned sub-element hierarchy
The leaf will arguably ALWAYS be considered for the 'dc' keywords even for DxO PLS.2.0 option but use the parameters anyway
(Conditional on Parameter #3 - typically All or Last) or (Leaf):
ADD ['Value' from 'Keywords'] TO 'dc Keywords'
(End-Conditional)
...
** Optionally add ['Value' from 'Keywords'] to 'hr' Keywords, i.e. add hierarchical keyword components to 'hr'!?
Unlike except for the "First" Hierarchical keyword component but simply do as the parameters indicate
(Conditional on Parameter #4 - typically F(irst) but arguably included in AC coding)
IF 'Parameter #4' = "A"
ADD ['Value' from 'Keywords'] TO 'hr Keywords'
END_IF
(End-Conditional)
...
Using the 'ParentId' access the linked keyword in the 'Keywords' Table
FOR each entry in 'Keywords' Table found using the 'ParentId' from a 'Keywords' Table entry
ACCESS 'Keywords' VIA ['ParentId' of 'Keywords'] Table entry
IF 'ParentId' = NULL Then
** This is the head or F(irst) keyword of the hierarchical keyword, i.e. A of A|B|C|D
PROCESS HEAD OF AN HIERARCHICAL KEYWORD
** Add keyword to the front of a reconstruction of the hierarchical keyword from leaf to top, i.e. in front of the previous k
ADD ['Value' from 'Keywords'] TO front of 'Hierarchical keyword'
** Reconstruction of hierarchical keyword is now complete and the head node should be added to the 'hr' fields of the image
This will occur in both the A (All) and AC (All combinations) cases
(Conditional on Parameter #6):
IF 'Parameter #6' = "A"
OR 'Parameter #6' = "AC" THEN
ADD 'Hierarchical keyword' TO 'hr Keywords'
END_IF
(End-Conditional)
...
and optionally add the re-constructed keyword to the 'dc' fields of the image
only really for non "hr" aware software e.g. ACDSee, ExifPro, Zoner
(Conditional on Parameter #5):
IF 'Parameter #5' = "A" THEN
ADD 'Hierarchical keyword' TO 'dc Keywords'
END_IF
(End Conditional)
...
Optionally add the keyword to the 'dc' keys this is different between pre PLS.2.0 & post PLS.2.0
(Conditional on parameter #3):
IF 'Parameter #3' = "A" THEN
ADD ['Value' from 'Keywords'] TO 'dc Keywords'
END_IF
(End Conditional)
...
Add keyword to 'hr Keywords', typically done with the head of an hierarchical keyword
(Conditional on Parameter #4):
IF 'Parameter #4' = "A"
OR 'Parameter #4' = "F" THEN
ADD ['Value' from 'Keywords'] TO 'hr Keywords'
END_IF
(End Conditional)
ELSE
** This is an intermediate hierarchical keyword
PROCESS INTERMEDIATE HIERARCHICAL KEYWORD
ADD |["Value' from 'Keywords'] TO front of 'Hierarchical keyword'
ADD |['Value' from 'Keywords'] TO front of 'Hierarchical keyword'
...
Optionally add the keyword to the 'dc' keys - this is different between pre PLS.2.0 & post PLS.2.0
```

```
(conditional):
ADD 'Value' from 'Keywords' TO 'dc Keys'
(End Conditional)
** Add 'Value' from 'Keywords' to 'hr' Keywords, typically done with the head of a hierarchical keyword
** but not necessarily with other keywords
(conditional):
ADD 'Value' from 'Keywords' TO 'hr Keywords'
(end Conditional)

END-IF
REPEAT-FOR UNTIL 'ParentId = NULL'
END-IF
REPEAT-FOR UNTIL 'NOTFOUND' IN 'ItemsKeywords'
** Arguably the 'dc' keywords & 'hr' keywords need duplicate entries resolved (de-duping), I believe

114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211

(Conditional on Parameter #3):
IF 'Parameter #3' = "A" THEN
  Capture All 'hr' keywords components in 'dc' keywords
  ADD ['Value' from 'Keywords'] TO 'dc Keywords'
  END-IF
  (End Conditional)
  Add 'Value' from 'Keywords' to 'hr' Keywords, typically done with the head of a hierarchical keyword
  but not necessarily with other keywords but that is the value of using parameter driven,
  "just" do what the parameters indicate
(Conditional on Parameter #4):
IF 'Parameter #4' = "A" THEN
  Capture 'hr' keyword components in 'hr' keywords (unlikely except for first but do what the parameters dictate)
  ADD ['Value' from 'Keywords'] TO 'hr Keywords'
  END-IF
  (End Conditional)
  IF Parameter #6 = AC then all combinations above this keyword need to be constructed, e.g. A|B|C or C>B>A
  IF 'Parameter #6' = "AC" THEN
    This is the point where a check could/should be made to determine if the current item is already assigned
    READ 'ItemsKeywords' with 'ItemId' = 'Current ItemId' AND 'KeywordsId' = ParentId
    IF NOTFOUND THEN
      'AC' must be satisfied programmatically because it won't happen via assignment
    Reconstruct an hierarchical keyword excluding the current item
    ADD Intermediate 'Hierarchical keyword' TO 'hr Keywords'
    END-IF ('Value'...)
    END-IF ('Parameter #6 ...')
  END-IF
  REPEAT-FOR UNTIL 'ParentId = NULL'
  END-IF('ParentId' = 'NULL' THEN)
  REPEAT-FOR UNTIL 'NOTFOUND' IN 'ItemsKeywords'
  ** Arguably the 'dc' keywords & 'hr' keywords need duplicate entries resolved (de-duping), I believe
```