



האוניברסיטה
העברית
בירושלים
THE HEBREW
UNIVERSITY
OF JERUSALEM

Circumnavigating NR syntax limitations

Hanoch Roniger

Mt. Scopus Library for the Humanities and Social Sciences

hanochr@savion.huji.ac.il

2021-05-20

Topic list

- [Overview](#)
- [Complex *if* conditionals](#)
- [Check string length](#)
- [Capitalize first character](#)
- [Problematic functions](#)
- [Manipulating repeatable fields](#)
- [Mathematical functions](#)
- [Position sensitive functions](#)
- [Conclusion](#)



Overview

- The NR syntax is very limited and does not include many features that we would have liked such as variables, ability to call functions, loops, etc.
- The purpose of this presentation is to illuminate some limitations of the NR syntax and to offer various workflows I have created to circumnavigate these limitations.



Complex *if* conditionals

- *when* conditions can have boolean operators but are checked only at the start - they will be true or false based on the state of the record before any manipulation regardless of its current state.
- *if* conditions can only have *or* operators in very specific situations - when the condition checks a string value in a subfield. They are checked only when they are called, ie. they are affected by prior manipulation of the data by the NR.



Complex *if* conditionals

- My solution: use a temporary subfield as a variable.
- Example: removing fields that are missing both ‡a & ‡z

```
addSubfield "035.Z.Z" if (exists "035.a.*")
addSubfield "035.Z.Z" if (exists "035.z.*")
removeField "035" if (not exists "035.Z")
removeSubField "035.Z"
```

or

```
addSubfield "024.X.X"
suffix "024.X" with "noa" if (not exists "024.a")
suffix "024.X" with "noz" if (not exists "024.z")
removeField "024" if (exists "024.Z.Xnoanoz")
removeSubField "024.X"
```



Check string length

- Check string length - 2 options:
 1. if possible characters are a known and closed list, mark border after each character and split:

```
replaceContents "TMP.a.0" with "0__"  
replaceContents "TMP.a.1" with "1__"  
[...]  
splitSubField "TMP.a.__" to "TMA.a" addSeq "S"  
<function> if (exists "TMA.S.<max_number>")  
<function> if (exists "TMA.S.<max_number - 1>")  
[...]
```

1. if list all possible characters is too long/unknown, convert subfield to control field and check by position:

```
copyControlField "TMP.a" to "002"  
<function> if (existsControl "002.{7,1}.-")  
<function> if (existsControl "002.{8,1}.-")
```



Capitalize first character

- Mark start of subfield and replace contents:

```
prefix "TMP.a" with "___"  
replaceContents "TMP.a.___a" with "A"  
replaceContents "TMP.a.___b" with "B"  
[...]
```

- I would advise replacing legacy characters with letters + combining diacritics to reduce the length of the code and reduce redundancy:

```
replaceContents "880.*.Ā" with "Ā" # U+0100 > U+0041 +  
U+0304  
replaceContents "880.*.ā" with "ā" # U+0101 > U+0061 +  
U+0304
```



Problematic functions

- The functions *changeSubFieldExceptFirst* & *correctDuplicateSubfields* do not behave the same way as other similar functions.
- Running these function of fields that have undergone manipulation by the same NR may result in unexpected behaviors.
- Both functions seems to use a cached version of the field when deciding which subfields are changed.
- Open case: #00919834 - Pending Development
- We were told that redesigning these functions' behavior to be consistent will take a long time.



Problematic function - *changeSubFieldExceptFirst*

```
035 ## $$a 1 $$a 2 $$a 3
```

```
changeSubFieldExceptFirst "035.a" to "b"
```

```
changeSubFieldExceptFirst "035.b" to "c"
```

```
↪ 035 ## $$a 1 $$c 2 $$c 3
```

- This NR results in all subfields except the first to change every time it runs.



Problematic function - *changeSubFieldExceptFirst*

- My solution: Using `changeSubFieldOnlyFirst` instead

```
035 ## $$a 1 $$a 2 $$a 3
```

```
changeSubFieldOnlyFirst "035.a" to "A"
```

```
changeSubFieldOnlyFirst "035.a" to "b"
```

```
changeSubFieldOnlyFirst "035.a" to "c"
```

```
changeSubField "035.A" to "a"
```

```
changeSubField "035.b" to "b" # reorder
```

```
changeSubField "035.c" to "c" # reorder
```

```
↳ 035 ## $$a 1 $$b 2 $$c 3
```



Problematic function - *correctDuplicateSubfields*

```
035 ## $$a (0CoLC)1121422924 $$z (0CoLC)1121422924
```

```
changeSubField "035.z" to "a"
```

```
correctDuplicateSubfields "035" "a"
```

↪ 035 ##

- This NR results in all subfields being deleted and leaves a blank field that doesn't even have a blank subfield.



Problematic function - *correctDuplicateSubfields*

- My solution: Separate into separate fields, use `correctDuplicateFields` and combine the fields again

```
035 ## $$a (0CoLC)1121422924 $$z (0CoLC)1121422924
```

```
changeField "034" to "TMP" if (exists "034.*.(0CoLC)*")
suffix "TMP.*" with "___"
changeSubField "TMP.z" to "a"
splitSubField "TMP.a.___" to "034.a"
correctDuplicateFields "034"
combineFields "034" excluding ""
removeField "TMP"
changeSubFieldExceptFirst "034.a" to "z"
```

```
↪ 035 ## (0CoLC)1121422924
```



Manipulating repeatable fields: combine & extract

- When calling functions, they affect all fields that fit the run on all existing fields, there are no functions that can be run only affect a single occurrence
- My solution:
 1. Add placeholder subfields in case a subfield is missing
 2. Combine fields
 3. Extract one field at a time by using:
copyField "<tag>.<subfield>" + CombineFields
or
addSubField + replaceSubFieldContents
 4. Use changeSubFieldOnlyFirst to get rid of the each extracted subfield after extraction
 5. Remove any temporary fields



Mathematical functions

- The NR syntax does not have mathematical functions.
- some basic calculations can be done by replacing numbers with text, manipulating it and replacing it back with numbers
- Examples of checksum calculations:
 1. ISBN 10 to 13
Proof of concept - only processes a single instance
checksum=(10-(mod 10(1 x odd digits + 3 x even digits))
https://drive.google.com/file/d/1iq_n2kKhfBnLqE9aLupmaEa-YuHpPrs4/view
 1. Danacode: short to 12 digits
Includes my combine & extract workflow; can process up to 4 instances per run
checksum=(mod 10(7 x odd digits + 9 x even digits))
<https://drive.google.com/file/d/17URFQL0m1XA0N9BlzF3XfNNm-XAzWjfz/view>



Position sensitive functions - ISBD punctuation

- The NR syntax does not have any functions that take into account the order in which the subfields appear in a given field.
- ISBD punctuation is dictated by the following subfield (space+colon before ‡b, space+slash before ‡c, etc.)
- We asked for such a function via ideas exchange:
<https://ideas.exlibrisgroup.com/forums/308173/suggestions/37313869>
- It is also in NERS request #6908



ISBD punctuation - non repeatable fields

- I found a way of creating a temporary field containing a string with the order of subfields using this workflow:
 1. copy the field to a temporary field
 2. mark each subfield code within the string
 3. break field into separate fields with a sequence number
 4. suffix the subfield codes to a temporary subfield sequentially
 5. clean up temporary fields & subfields
- ISBD punctuation in 245 (20210504)
<https://drive.google.com/file/d/12rR7p155WY079XN2ISTpLqEs43C-GQnO/view>



ISBD punctuation - repeatable fields

- Using the same in combination with with my combine & extract workflow I was able to do this for repeatable fields as well.
- Note: the extraction needs to run n times, n being the maximum total number of individual subfields in a given record. This may cause the NR to be extremely long.
- ISBD punctuation in 246 (20210509)
<https://drive.google.com/file/d/1iCW633IizB6KUPGZIB51zbrfSiUkSLR9/view?usp=sharing>



Conclusion

- With my latest breakthrough it is possible to manipulate values based on their relative position.
- In conjunction with other workflows shown here, we now have enough tools to make most of our necessary manipulations using NRs.
- These workarounds require much more code, so they are not a replacement for the development of more concise functions by Ex-Libris.
- Unfortunately we were notified that a fix for functions' inconsistent behavior and development of missing functions would take a very long time.
- A collection of my NRs are open to the public on my google drive: <https://drive.google.com/drive/folders/1DW1EI0cEYaUXusdDCKoV--liyQ3dTHo8?usp=sharing>

