

Assigning selections to headings

We start with a document that has some paragraphs marked as ‘headings’ in some form. Other paragraphs are our selections. We want to know for each selection, under which heading it belongs, so we can process it in some relevant way (give it a marker, etc.).

We will start by finding each type using a relevant find expression

```
$headings = $doc.text.find ...  
$sels = $doc.text.find ...
```

Now the idea is to cycle through the selections while keeping track of the most recent **heading**. We will use a pointer (**\$i**) as the index of the most recent **heading**.

As we loop we will need to check:

- *Is there* a next **heading**
AND
- is *that next heading** located before the current selection

This check needs to be done in a loop which repeats until we find a suitable **heading**.

The problem is that the condition cannot be checked all at once, since the second part will throw an error¹ if there is no next **heading**, but the condition must be revisited in a loop until we find a suitable **heading**.

We’ll consider 3 possible approaches

Approach 1:

/ This approach uses an endless loop and checks the condition inside the loop.*

*It uses 'break' to escape the loop. The downside is that it is ugly. */*

```
$i = 0  
foreach $sel in $sels  
  while @true  
    if ($i+1) < $headings.count  
      if $headings[$i+1].location < $sel.location  
        $i += 1  
      else  
        break  
      end  
    else  
      break  
    end  
  end  
  # $headings[$i] is the most recent heading for $sel  
end
```

¹ Programmers familiar with languages like Perl will know that in such languages this can be checked without a problem, since those languages will not test the second condition, if the first fails. But unfortunately Nisus Macro Language is not such a language.

Approach 2:

/* This approach makes clear that the location check is the condition for the loop.

The downside is that we need to write the limit check twice, before and again inside the loop.

*/

```
$i = 0
foreach $sel in $sels
    if ($i+1) < $headings.count
        while $headings[$i+1].location < $sel.location
            $i += 1
            if ($i+1) >= $headings.count
                break
            end
        end
    end
    # $headings[$i] is the most recent heading for $sel
end
```

Approach 3:

/* This approach uses a defined command to create a single condition for use in the loop */

```
Define Command NextHeadingBeforeLoc ($headings, $next, $loc)
    if $next < $headings.count
        if $headings[$next].location < $loc
            return @true
        end
    end
    return @false
end

$i = 0
foreach $sel in $sels
    while NextHeadingBeforeLoc ($headings, $i+1, $sel.location)
        $i += 1
    end
    # $headings[$i] is the most recent heading for $sel
end
```