`Utility Provider.ContextValue(1)` is "2\two"

`Utility Provider.ContextValue(2)` is "1\one"

`UtilityProvider.ContextValueCount` is 3, the number of list arguments.

The value of `displayString` starts at "", then is "B", then "B2", then "B21" as the loop is executed. "B21" is what is sent to the application.

As the lists get more complex, this basic command can do more complex operations. Or, for some applications requiring special formatting, the loop can be modified slightly.

For instance, consider an application requiring heavy outlining in a specific style. The basic outline form is:

<roman_numeral>.<uppercaseAlpha>.<digitperiod>.<lowercaseAlpha> <lowercaseRoman>.

and the upper-level outline elements are:

<roman_numeral>.<uppercaseAlpha>.<digitperiod>.<lowercaseAlpha>

<roman_numeral>.<uppercaseAlpha>.<digitperiod>.

<roman_numeral>.<uppercaseAlpha>.

<roman_numeral>.

For this example, a sample format is III.B.3(d)(iv), to be dictated “three B. three D. four”. For clarity, the user will say the key word “outline” prior to the 1-5 elements.

By creating the lists appropriately, the punctuation need not be spoken at all!

```
<roman_numeral>
I.\one
II.\two
…
<uppercaseAlpha>
A.\A.
B.\B.
…
<digitperiod>
1.\one
2.\two
…
<lowercaseAlpha>
(a)\A.
(b)\B.
…
<lowercaseRoman>
(i)\one
(ii)\two
```