Counter styles

You can define your own unique counter style with ‘@counter-style’. Specify the counter-style name, the counter-system type in ‘system’, and the counter-symbol in ‘symbols’. Names specified in the counter-style rule can be used in the ‘list-style-type’ property in CSS. The ‘@counter-style’ rule must be placed in the CSS stylesheet. You may also use styles defined in Predefined Counter Styles, including styles in Section 7, Complex Predefined Counter Styles. For further details on CSS Counter Styles Level 3, please refer to the W3C specification. Counter style settings are available in Antenna House Formatter V6.3 and later.

These samples, with some exceptions, convert the numerals in the character string `One: 1, Two: 2, Three: 3, Four: 4, Five: 5, Six: 6` by using the ‘counter()’ function with a counter style defined in a ‘@counter-style’ rule. The numbers to be converted are in red.

Custom Counter Styles

```
system: cyclic; symbols: a b c;
```

Converts the numerals in the character string into the repeating sequence a, b, c, a, b, c ...

One: a , Two: b , Three: c , Four: a , Five: b , Six: c

```
system: numeric; symbols: a b c;
```

The first symbol, ‘a’, corresponds to 0, so the converted numerals start from ‘b’.

One: b , Two: c , Three: ba , Four: bb , Five: bc , Six: ca

```
system: alphabetic; symbols: a b c;
```

Converts the numerals in the character string into the sequence a, b, c, aa, ab, ac, ba, bb, bc, ...

One: a , Two: b , Three: c , Four: aa , Five: ab , Six: ac

```
system: symbolic; symbols: a b c;
```

Converts the numerals in the character string into the sequence a, b, c, aa, bb, cc, aaa, bbb, ccc, ...

One: a , Two: b , Three: c , Four: aa , Five: bb , Six: cc

```
system: additive; additive-symbols: 5 v,1 i;
```

Converts the numerals in the character string by converting 5 to v and remainder multiples of 1 to the same number of i.

One: i , Two: ii , Three: iii , Four: iiii , Five: v , Six: vi

```
system: fixed; symbols: a b c;
```

Converts the numerals in the character string into the sequence a, b, c and then displays higher numbers unchanged.

One: a , Two: b , Three: c , Four: 4 , Five: 5 , Six: 6

```
system: extends decimal; pad: 2 '0';
```

‘pad’ specifies padding for short representations. This displays up to two digits in the sequence 01, 02, 03, ...

One: 01 , Two: 02 , Three: 03 , Four: 04 , Five: 05 , Six: 06

```
system: cyclic; symbols: a b c; range: 1 3; fallback:
cjk-decimal;
```

‘range’ specifies the number range to which the style applies. ‘fallback’ specifies the fallback style for numbers outside that range. This sample applies a, b, c for numerals 1 to 3. It then falls back to using the ‘cjk-decimal’ style from Predefined Counter Styles for 4 and above.

One: a , Two: b , Three: c , Four: 四 , Five: 五 , Six: 六

```
system: extends decimal; negative: "[-" "]";
```

‘negative’ specifies the prefix and suffix character strings for negative values. This sample displays `[-` before, and `]` after, negative values.

Negative One: [-1] , Negative Two: [-2] , Negative Three: [-3] , Negative Four: [-4] , Negative Five: [-5] , Negative Six: [-6]

Complex Predefined Counter Styles

Examples from Section 7, Complex Predefined Counter Styles, of Predefined Counter Styles.

‘circled-decimal’

One: ① , Two: ② , Three: ③ , Four: ④ , Five: ⑤ , Six: ⑥

‘filled-circled-decimal’

One: ❶ , Two: ❷ , Three: ❸ , Four: ❹ , Five: ❺ , Six: ❻

‘fullwidth-upper-alpha’

One: Ａ , Two: Ｂ , Three: Ｃ , Four: Ｄ , Five: Ｅ , Six: Ｆ

‘lower-greek’

One: α , Two: β , Three: γ , Four: δ , Five: ε , Six: ζ

‘japanese-informal’

One: 一 , Two: 二 , Three: 三 , Four: 四 , Five: 五 , Six: 六

‘japanese-formal’

One: 壱 , Two: 弐 , Three: 参 , Four: 四 , Five: 伍 , Six: 六