

WORKING WITH SEQUENCES

Defining Part Numbers

In order to include a sequence in the part number, an administrator can use one of three (3) tags. They are:

PSEQ – product sequence
GSEQ – global sequence
SSEQ – nested sequences

PSEQ is a product sequence, which allows several products to use the same sequence independently of each other. In other words, under this scenario, two products can use the same sequence in their part numbers. For example:

Product ABC part numbers can be: ABC0001, ABC0002,
ABC0003...
Product XYZ part numbers can be: XYZ0001, XYZ0002,
XYZ0003...

Both products have the same sequences – 0001, 0002, 0003,...

GSEQ is a global sequence, which allows several products to share the same sequence. In other words, under this scenario, two products increment the same sequence and therefore a sequence number is only used once. For example:

Product ABC part numbers can be: ABC0001, ABC0002,
ABC0004...
Product XYZ part numbers can be: XYZ0003, XYZ0005,
XYZ0006...

Here Product ABC uses the sequence numbers 0001 and 0002. Product XYZ used the sequence number 0003. Since 0003 had already been used the next sequence number available for Product ABC was 0004.

SSEQ allows sequences to be nested. For example, two sequences go from 1 to 3 and they are named sequence One and Two. The expression <*SSEQ (one, two)*> would result in 11, 12, 13, 21, 22, 23, 31, 32, 33.

Another example would be to have one sequence name of "ABC" with string elements A, B, C, and another sequence named "Ten" going from 1 to 10, then expression `<*SSEQ(ABC, Ten)*>` would result in A1, A2, A3, A4, A5, A6..., A9, A10, B1, B2, ..., B10, C1..., C10.

Similarly, if a portion of a part number is defined as `<*SSEQ(Ten, ABC)*>` then the results would be 1A, 1B, 1C, 2A, 2B, ...10C.

Syntax

The actual syntax for using the above tags is `<*PSEQ(sequence name)*>`, `<*GSEQ(sequence name)*>`, `<*SSEQ(sequence name1, sequence name 2, ..., sequence name n)*>`. For more information on syntax see "Tabs and Expressions."

There are no limits as to how many sequences can be used in defining a part number. For example a part number can be defined as `<*Value(Number of Poles)*><PSEQ(some seqname)*><*Value(another attribute)*><PSEQ(another seqname)*><*GSEQ(third sequence)*><*SSEQ(seq four, seq five)*>`.

Conditional Sequences

Sequences can conditionally be included in the part number. This is accomplished by using the familiar Microsoft "If" syntax which is:

```
[If](condition) {expression if true}{expression if false – note this is optional} [ENDIF]
```

The condition itself uses the following syntax `([operator](variable1, variable 2))`.

For example if there is an attribute called Number of Poles and if that needs to use the numbers from 2000 to 2999 from a sequence named "2000-2999" when the number of poles is "2", but use numbers from 3000 to 4999, from a sequence named "3000-4999", when the number of poles is greater than 3, the expression constructing the part number would be as follows:

```
<*Value(some attribute)*><*CatCode(another attribute)*>  
[IF]([EQ](<*Value(Number of Poles)*>, 2)){PSeq( 2000-2999)}[ENDIF]  
[IF]([GT](<*Value(Number of Poles)*>, 3)){PSeq( 3000-4999)}[ENDIF]
```