

Managing Complex Autonumbering Sequences in Adobe FrameMaker®

Lester C. Smalley
lsmalley@infocon.com
Information Consultants, Inc.
P.O.Box 310
662 Yorklyn Road
Hockessin, DE 19707

Supplemental information prepared exclusively for
INFOCON customers

1.0 Purpose

FrameMaker supports an extremely sophisticated and complex automatic numbering capability. While there is a wide variety of “pre-built” templates supplied with the program (or available via the internet such as from [ADOBE’S WEBSITE](#)) that incorporate – to varying degrees – paragraph formats with autonumbering, many users do not take more than token advantage of the available features. In part, this may be due to a lack of need of these writers, but may also be because Frame’s autonumbering function is documented in a dense and somewhat obscure manner that makes it difficult to fully understand its power and potential uses.

The purpose of this report is to help clarify some of the capabilities of the autonumbering feature, as well as illustrate one possible complex numbering scheme. This example is intended for use in a large, multi-file book project and, once setup, requires essentially no intervention from the author, yet permits numbering to cascade across chapters, incrementing the chapter number, four levels of headings, figures, tables, examples, and two levels of lists.

The base assumption underlying this example is the requirement for a numbering scheme that is based on the chapter number as the major level indicator. That is, in Chapter 3, the first figure will be Figure 3-1; the first table will be Table 3-1, the first heading will be 3.1, and so on. While other numbering systems are certainly possible (such as figures and tables numbering in a simply ascending series, but headings restart numbering in each chapter), this report does not explicitly address the format options needed to create these. Furthermore, it is presumed that individual chapters are able to be saved in a single file, and thus this paper does not discuss managing the page numbering scheme driven by the chapter number (i.e., page 3-27) across multiple files.

Hopefully, the reader will gain a solid foundation in the capabilities of FrameMaker’s autonumbering function and be able to create and manage such alternate series after reading this document. The key to success is to experiment and practice, but this paper should provide a bit of a head start over trying to master the information in the Frame documentation alone.

2.0 Numbering Properties

FrameMaker associates numbering with specific paragraph formats, as usually only selected elements in a document require being counted. However, the actual numeric evaluation is done, not based on a given paragraph tag but on a specific counter which is independent of any particular format. This counter may be referenced by multiple paragraph tags. For example, the generic “Blank Paper” templates (provided with the program) use a pair of tags to start (or restart) and to increment numbered lists. These tags, *Numbered1* and *Numbered*, both reference the same counter. *Numbered1* always sets the counter to one; *Numbered* adds one to the counter.

2.1 Autonumber Formats and Counters

There are a variety of “building blocks” available in a scrolling list within the Numbering properties of the paragraph designer. These affect the way the counter is displayed and how the counter may change, but do not affect *which counter* is being used internally to keep track of a value. If it is not explicitly identified, you will always be referencing the default counter and will have only one series of numbers in a document, regardless that you may reset it, change it to an arbitrary value or display it as an arabic or roman numeral, or a letter.

A simple description of the autonumber information, ignoring for the moment fixed information (such as a bullet or tab character) or user entered text (the word *Figure*, for example) is:

Series_Label:<Format Option>

The **Series_Label** must be a single letter (upper- or lower-case) followed by a colon *as the very first two characters* entered in the edit box of the numbering properties dialog. A series label is used to identify a different sequence of numbering in a document, and designate a different internal counter to manage them. The series label can be omitted, in which case you are using the default (or unlabeled) counter. Thus, there are 53 (26 uppercase + 26 lowercase letters + 1 unlabeled) different numbering series possible in a FrameMaker document.

The **Format** is a single character designating how the counter will be displayed. The choices are:

- n** – Arabic numerals;
- a** or **A** – lower- or uppercase alphabetic characters;
- r** or **R** – lower- or uppercase Roman numerals;
- space** – do not display the value.

The **Option** controls how the counter changes when this paragraph is used in the document. Valid option choices are:

- null** – do not change the value;
- +** – add one to the counter (FrameMaker can only increment counters by one);
- =Val** – set the counter to ‘Val’ (default is 1, but may be changed to any value)

The value of a counter is changed based on the **option before** the **format** is checked to display the value. The characters ‘<’ and ‘>’ are required to enclose the format and value options, even if they are a space and null.

It is possible to have a building block that is simply “< >” (note the space between the delimiters) which means, do not display the value and do not change the value. You can also have building blocks that set a counter but do not display a value, such as **< =0>**. We will illustrate shortly how these are placeholder blocks used, and how useful such ‘unorthodox’ counter elements can be. In fact, the empty ‘< >’ placeholder is listed at the very end of the autonumbering building blocks scroll list (found under the Numbering Properties in the Paragraph Designer window).

2.2 Compound numbering

If you compare the numbering properties of the tags *Numbered1* and *Numbered*, you will see that only one of the counter building blocks is present in either tag. In the case of *Numbered1*, this is the `<n=1>` construct, which translates to “display the value as an Arabic numeral after setting the counter to one.” *Numbered* uses the `<n+>` building block, which simply means “display the value as an Arabic numeral after adding one to the counter.”

You may use more than one building block in a given format to create multilevel or compound numbers, such as Table 3-1 or Section 3.5. In fact, complex numbering up to 16 levels deep are possible in FrameMaker. This is accomplished by using several building blocks at a time in the numbering properties format. FrameMaker assigns the counters decreasing significance from left to right.

A more accurate description of the autonumber information from above is:

Series_Label:[Optional Text]<Format Option>[Optional Text<Format Option>...]

The optional text is frequently used so that the autonumber string will include a specific word, such as **Chapter** or **Figure**, or to provide punctuation between the counters (such as a dash or period) in its output whenever the user selects the specific paragraph tag.

The **[Optional Text<Format Option>...]** indicates that we may repeat this grouping and thus can include multiple building blocks in our autonumber string, to create a hierarchy of numbers. In fact, you can use up to 16 counters in a single autonumber format, and at each level these can use different format choices to display the counter, so you might have a style that displays as “Section 2-C”. To create numbers such as are used for the section headings in this report, we need two counters, one each for the major and minor levels.

We might define a pair of tags “1Heading” and “2Heading” with autonumber definitions of `<n+>.<n=0>` and `<n>.<n+>` respectively. The 1Heading tag increments the first level counter and always sets and displays the second level counter as zero. And the 2Heading tag keeps the first level counter unchanged, and increments the second level only. The understanding here is that any time a 1Heading paragraph is used, it will reset the sublevel; and any following 2 Heading is a sublevel of that heading, and should have the same major number. The headings in this report are setup this way.

2.3 Resetting counters

FrameMaker provides an explicit building block to reset a counter, or optionally, to set it to a specific value. However, we wish to create a numbering system that is automatic, in that it will correctly set the value needed based on the sequence of paragraphs in the document. In addition to the forced reset provided by the “=Num” option, FrameMaker uses an implied reset feature: whenever a paragraph contains a numbering format that *does not* include any building blocks for sublevels (either an explicit choice or a placeholder), the sublevel counters *are automatically reset to zero*.

This means we could set up the following tags and numbering properties definitions which will reset and increment automatically to provide three levels of numbered headings in our document:

Heading1	<code><n+>.</code>
Heading2	<code><n>.<n+>.</code>
Heading3	<code><n>.<n>.<n+>.</code>

Notice that each “more significant” heading does not contain any building block for any lower levels. This utilizes the “auto-reset” feature for those lower levels. Whenever the Heading1 format is used, it automatically resets both the second and third level counters to the default value of zero.

2.4 Holding a value without displaying it

In a complex numbering scheme, we may have a number of different items to track. And we do not want to see or change the values of other items inadvertently. We need a way to ensure that we have the right table number, but that it does not affect numbering for heading levels, or figures. This can be accomplished if we create separate series of numbering, so that figures, tables, lists, and headings are all completely independent. The problem with this approach, however, is that we either need to create special “start” tags to assign the correct initial value for the element, or we need a single series in the document constructed so that it can have some items not affect other types.

Earlier, an empty building block was described that did not display a value or alter the counter used ([SEE SECTION 2.1 ON PAGE 2](#)). The < > in the scroll list is a place holder, which becomes particularly important if there are multiple building blocks being used to create a compound numbering system, such as Chapter 3, Section 3.1, and Table 3-1. This feature is used extensively in the sample below.

3.0 A Sample Configuration

In this example, a number series is defined that incorporates eight levels of items that are linked together. Sample paragraph tag names, autonumber definitions and generated output are:

TAG	NUMBERING PROPERTIES	SAMPLE RESULT *
Chapter	H:Chapter <\$chapnum>	Chapter 5
Figure	H:Figure <\$chapnum>-<n+>< >< >< >< >< >	Figure 5-1
Table	H:Table <\$chapnum>-< ><n+>< >< >< >< >	Table 5-1
Example	H:Example <\$chapnum>-< >< ><n+>< >< >< >	Example 5-1
1Head	H:<\$chapnum>< >< >>.<n+>	5.1
2Head	H:<\$chapnum>< >< >>.<n>.<n+>	5.1.1
3Head	H:<\$chapnum>< >< >>.<n>.<n>.<n+>	5.1.1.1
4Head	H:<\$chapnum>< >< >>.<n>.<n>.<n>.<n+>	5.1.1.1.1
Body:	B:< =0>	(reset steps but not display)
Step:	B:<n+>.\t	1. text of step
Sub-step:	B:< ><a+>.\t	a. text of sub-step
Bullet	\b\t (may also display as •\t)	• text of bullet

* The exact appearance requires setting indentation and tab stops under “Basic Properties” as well as the font characteristics under “Default Font Properties” in the Paragraph Designer for each of the tags.

In order of decreasing significance (that is, from left to right), the counter positions in this example are as follows:

- 1st: chapter number (defined by the chapter number system variable¹)
- 2nd: figure number
- 3rd: table number
- 4th: example number
- 5th: first-level heading number
- 6th: second-level heading number
- 7th: third-level heading number
- 8th: fourth-level heading number

1. This is new in version 6.0. If you are using an older version, replace <\$chapnum> with <n+> in the “Chapter” format example and with the <n> building block in the other examples above.

This entire sequence is carefully structured to produce the desired effect of properly incrementing items throughout the document:

- the chapter number resets all sublevel counters for tables, figures, examples, and headings;
- the figure numbers, table numbers, and example numbers are invoked in the middle of the sequence because they don't have to be reset within a chapter;
- the heading numbers are invoked in the last few positions of the sequence because they do have to reset each other;
- the body style resets only the step and sub-step tags;
- the step and sub-step use a different counter than the other numbered items

The chapter number does not have to be set manually in each chapter², and all figures have the same style. The “Chapter” tag contains no special `< >` sequences, so all other dependent numbers are automatically reset at the start of a new chapter. Also the chapter number is defined as `<$chapnum>` so that it can be set automatically by continuous numbering across all chapters in a book³. Thus if you rearrange chapters in a book, the numbers are reset automatically. This could just as easily be hard-coded as `<n=5>` (for chapter 5) and would work. In this case, you would be able to see the correct chapter number in the file before it is put in a book, but the chapter would not be renumbered automatically if rearranged in a book—this would have to be done manually in the file and in all files that follow it in the book sequence.

However, if a figure number is changed, we don't want to reset the table, example, or any heading numbers. So, in the “Figure” tag, all other numbering positions are replaced with `< >`. The same is true for “Table” and “Example”.

For “1Head” through “4Head”, we must not reset the figure/table/example numbers, and so for those three positions use the `< >` placeholder building block. However, every “1Head” should reset the 2nd, 3rd, and 4th heading levels, so there is nothing after the `<n+>`. The same is true for the “2Head” through “4Head” paragraphs. Using this example system, you would get a numbered sequence like:

5.1 *1Head*

5.1.1 *2Head*

5.1.1.1 *3Head*

5.1.1.1.1 *4Head*

5.1.2 *2Head*

5.2 *1Head*

5.2.1 *2Head*

etc.

If you do not number headings in your document, you can still keep the coding in the tag – turned off in the Paragraph Designer – making it easy to enable if you ever do want numbering.

-
2. Changing the chapter numbers is accomplished automatically by the FrameMaker book file when each individual file (a chapter) has been setup to continue paragraph numbering from the previous file.

For complete details on creating and managing multifile books, refer to the standard FrameMaker documentation (for version 5, *Using Adobe FrameMaker*, Chapter 24 [BOOK BUILDING](#); in version 5.5, *Adobe FrameMaker User Guide*, Chapter 14 [BOOKS](#); in version 6, *Adobe FrameMaker 6.0 User Guide*, Chapter 14 [BOOKS](#).)

3. Because the initial value of any counter is zero, the first time the `<n+>` building block is encountered it will display the value as 1.

Finally, the steps are defined in a separate series. In this example, the Body style resets both Step and Sub-step numbers without displaying either counter, and the Step tag resets the sub-steps. The two assumptions being that the ‘steps’ are unrelated to the major numbering sequence for the chapter, but it is quite possible that you will need to have more than one set of steps within any numbered section of the file; and that you will have at least one ‘Body’ paragraph between sets of steps. Thus you can get:

Body text.

1. A step
 - a. A sub-step
 - b. A sub-step
2. Another step

Some more Body text.

1. Next series of steps...

This avoids the necessity of a separate tag for the first step which has the sole purpose of resetting the counter to 1, and a different tag for the additional steps in a series that increments the counter. While not incorrect, a system that requires the writer to force resetting values makes errors a bit more likely, due to the possibility of incorrectly tagging an item or from cut and paste operations to rearrange a sequence of steps. The system above however, assumes at least one Body paragraph exists between sets of steps so the counter is reset, which means it is not bulletproof either, if the “separator” paragraph is marked with a different tag, or only a heading is used between sets of steps.

4.0 Numbering in FrameMaker Books

FrameMaker includes the ability to create multi-file documentation sets, called books. A book in Frame is simply a collection of normal FrameMaker document files, arranged in some particular order. A book may include generated files, such as a Table of Contents or an Index. The individual files in a book can be setup in various ways. Specifically, the book permits the control of the starting page side for each file, whether page numbering continues from the previous file or restarts in a file, and whether the paragraph numbering continues or restarts in a file.

To support the example numbering scheme described earlier ([SEE SECTION 3.0 ON PAGE 4](#)), the one requirement is that in the book file, each chapter is setup to continue chapter, paragraph and page numbering from the previous file, with the exception of the first chapter, which restarts from 1. If an index is included, it is also set to restart page numbering from 1. This allows there to be a separate page numbering sequence for the ‘front matter’, such as a Title page, Copyright notice, Table of Contents, Preface. In the “Format->Document->Numbering” dialog, the numbering style of the front matter files and index is set to use lower case Roman numerals, while the chapters are set to use standard Arabic numerals.

I typically number pages in a multi-file book project according to a common “user guide style” of 5-1, 5-2, etc. (chapter number–page number). In FrameMaker 6, this is encoded on the master pages as the “**Chapter number**” (<\$chapnum>) variable followed by a hyphen, followed by the “**Current Page #**” variable. Remember, from our example ([“A SAMPLE CONFIGURATION” ON PAGE 4](#)) series, that if you are using version 5.5.6 or older, you will have to substitute the <n> or <n+> building block for <\$chapnum> counter in all paragraph tags that need to display the chapter number.

One problem is that there is a **Page 1** in each chapter. In older versions of the program, there was no chapter number variable, so one of the four available running header/footer variables (**Running H/F 1**, for instance) was defined to extract the autonumber value of the chapter tag like so:

`<$autonumonly[Chapter]>` and used as the prefix to the Current Page # variable in the footer of the master page(s) in place of the chapter number variable.

Also in the Table of Contents (or any of the generated files like an index, or list of figures) the first page number extracted *from each file* will be simply shown as **1**. The solution to this possible dilemma also depends on the version of FrameMaker.

In version 6.0 and later, you simply need to add “`<$chapnum>`” and a separator (like a non-breaking hyphen) as a prefix to the `<$pagenum>` building block to the special Reference page flow that control the content of the generated file. For example, if you are creating a Table of Contents file, you will have a TOC reference page in it that has special paragraphs to govern the creation of the content in the table of contents.

If you are using any version prior to 6.0, the answer is found under the book file’s “File->Set up File” command; the “Page Number in All Generated Files” Prefix is set to use a hard coded value (i.e., the chapter number) and a dash (preferably, a non-breaking hyphen). This prefix must be manually setup for each file and update whenever files in the book are rearranged, added or deleted. Please note that this is completely unnecessary if page numbering is set to continue across chapters as each number is unique and there would be no confusion as to which page is being referenced.

The Running H/F variable is defined as `<$paranumonly[Chapter]>`⁴ so that it automatically picks up the chapter number from the Chapter tag. This is defined in the template used to create the individual chapters, so there is nothing the user needs to do to have this style used. This is also unnecessary if page numbers continue across the book. However, you might wish to have the whole phrase “**Chapter #**” appear as part of the header or footer. You can extract the proper information from the Chapter tag by defining the Running H/F variable as `<$paranum[Chapter]>` to display the entire autonumbered information, including any manually entered text, such as the word chapter, that was entered in the autonumber format.

5.0 Conclusion

Hopefully, this information helps clarify and expand on the description of FrameMaker’s numbering capabilities from the published manuals, and offers a sufficient number of examples and enough detail in the description to make the feature more accessible than the purely technical information provided in the standard product documentation. There is a tremendous degree of flexibility and customization possible, so that almost any numbering scheme imaginable can be implemented. Of course, any numbering system must be completely tested to be sure that it satisfies the requirements of the authors who will use it and the readers of the resulting documents.

4. This extracts the numeric value only of the paragraph tag Chapter, ignoring any prefix text (such as the word chapter), or trailing text (e.g., spaces or tab characters) that exist in the autonumber format.

Note: if you include text between building blocks in the autonumber definitions (such as a hyphen or periods to separate the chapter number and table number), these embedded characters are extracted even with the `<$paranumonly>` definition.

For complete details on running headers and footers, refer to the standard software documentation: for version 5, *Using Adobe FrameMaker*, Chapter 12 [VARIABLES](#) and Chapter 18 [BASIC PAGE DESIGN](#); in version 5.5, *Adobe FrameMaker User Guide*, Chapter 7 [VARIABLES](#) and Chapter 11 [PAGE LAYOUT](#); in version 6, *Adobe FrameMaker 6.0 User Guide*, Chapter 7 [VARIABLES](#) and Chapter 11 [PAGE LAYOUT](#).