FrameMaker Building Blocks

Using Building Blocks to Customize FrameMaker Documents

PO Box 12761
Research Triangle Park, NC 27709
sales@scriptorium.com
http://www.scriptorium.com

Building blocks are a very powerful feature of FrameMaker because they allow you to display information without having to type it. Like macros, they consist of code interpreted by FrameMaker, which then displays the appropriate information in your file. Building blocks are enclosed by angle brackets and are normally preceded by a dollar sign (<$year>, for example).

A few building blocks are used globally in FrameMaker. These global building blocks display information such as autonumbering, text, and paragraph tags. You see them most often in running headers and footers and cross-reference formats. Table 1 describes these building blocks.

Table 1: Global building blocks

Building block	Displays
<$paranum>	Paragraph's autonumber (including text)
<$paranumonly>	Paragraph's autonumber (excluding text)
<$paratag>	Paragraph tag's name
<$paratext>	Text of the specified paragraph

In addition, FrameMaker uses specific building blocks in the following features:

- Variables
- Autonumbering
- Cross-references
- Index markers
- Text flows

Variables

Variables use building blocks to display information such as month, date, year, time, file name, and page count. They also use building blocks to display running header/footer information such as headings and page numbers.

The default FrameMaker template includes several variables, two of which are Modification Date variables. The Modification Date (Short) variable displays the modification date of the document. This variable is defined by the building blocks <*$monthname*>/<*$daynum*>/<*$shortyear*>. The result looks like this:

9/20/99

Table 2 describes the building blocks used in this variable.

Table 2: Modification Date (Short) variable

Building block	Displays
<$monthnum>	Number of the month
<$daynum>	Number of the day
<$shortyear>	Year without the century

The Modification Date (Long) variable displays a more detailed date. This variable is defined by the building blocks *<$monthname>* *<$daynum>*, *<$year>* *<$hour>*:*<$minute>* *<$ampm>*. The result looks like this:

September 20, 1999 6:00 pm

Table 3 describes the building blocks used in this variable.

Table 3: Modification Date (Long) variable

Building block	Displays
<$monthname>	Name of the month
<$daynum>	Number of the day
<$year>	Year with the century
<$hour>	Hour
<$minute>	Minute
<$ampm>	Morning or evening designation

Other building blocks let you customize these variables. You can add the *<$second>* building block to include seconds. To display military time, use the *<$hour24>* building block instead of *<$hour>*. You can also display the minutes with a leading 0 using the *<$minute00>* building block. More building blocks are included in the FrameMaker documentation. Scriptorium Publishing's *FrameMaker Building Blocks Quick Reference* also lists building blocks. Download a copy at http://www.scriptorium.com/BBQuickRef.pdf.

Autonumbering

FrameMaker also uses building blocks in complex autonumbered steps and headings. These building blocks are called "counters." Counters designate how the numbered paragraph tags are incremented. For example, you can create a Step1 paragraph tag so your procedure steps begin with Step 1. This tag uses the <*n=1*> counter to start the numbering at 1. You can then create a Step paragraph tag for steps beyond Step 1. This tag uses the <*n+*> counter to increment the step's number by one. The result looks like this:

`Step1 ¶ tag` —— **1** I enjoy baking apple pie.
`Step ¶ tag` —— **2** My friends enjoy it, too.

You create the preceding autonumbering formats in the Numbering Properties of the Paragraph Designer. Table 4 describes the counters you can use.

Table 4: Numbering styles

Counter	Displays
<n>, <n+>, <n=1>	Numeric
<r>, <r+>, <r=1>	Lowercase roman numeral
<R>, <R+>, <R=1>	Uppercase roman numeral
<a>, <a+>, <a=1>	Lowercase alphabetic
<A>, <A+>, <A=1>	Uppercase alphabetic

In addition to single-numeral steps, you can create heading paragraph tags that display multiple numerals. For example, you may want your headings to look like this:

`Chapter ¶ tag` —— 1.0 At Home with Amanda Steubing
`Heading1 ¶ tag` —— 1.1 Baking apple pie
`Heading2 ¶ tag` —— 1.1.1 Making crust from scratch

To create these multinumeral headings, use the counters listed in Table 5 on page 5.

Table 5: Multinumeral headings

Paragraph tag	Counter	Displays
Chapter	<n+>.<n=0>	1.0
Heading1	<n>.<n+>	1.1
Heading2	<n>.<n>.<n+>	1.1.1

Cross-references

Building blocks provide the framework of cross-reference formats. They allow you to specify which pieces of information are displayed in the cross-reference.

The default FrameMaker template contains several cross-reference formats that you can customize. For example, the format called See Heading & Page displays the heading and page of the referenced paragraph tag:

See "Introduction" on page 4.

The cross-reference definition is *See "<$paratext>" on page <$pagenum>*. Table 6 describes this definition.

Table 6: Heading & Page cross-reference

Building block	Displays
<$paratext>	Paragraph text
<$pagenum>	Page number of the paragraph

If the chapters in your book are numbered, you can expand the cross-reference to include the chapter number:

See Chapter 1, "Introduction," on page 4.

The cross-reference definition is *See Chapter <$paranumonly>, "<$paratext>" on page <$pagenum>*. Table 7 describes this definition.

Table 7: Numbered Heading & Page cross-reference

Building block	Displays
<$paranumonly>	Paragraph's autonumber (excluding text and punctuation)
<$paratext>	Paragraph text

Table 7: Numbered Heading & Page cross-reference

Building block	Displays
<$pagenum>	Page number of the paragraph

The <*$paranumonly*> building block displays only the paragraph's autonumber. If you use the <*$paranum*> building block instead, the cross-reference displays all text and punctuation used in the heading's autonumbering format, which you want to avoid:

See Chapter 1:, "Introduction," on page 4.

Index markers

Using building blocks, you can specify single page entries and page range references in indexes. For example, suppose you want an index that looks like this:

A

apples
baking apple pie 11–13
types used in pies 11

P

pies. *See* apples.

The index marker that created these entries consists of the following text and building blocks:

<*$startrange*>apples: baking apple pie; <*$singlepage*>apples: types used in pies; <*$nopage*>pies. <*emphasis*>See <*Default Para Font*>apples.

Table 8 describes how FrameMaker reads these building blocks.

Table 8: Index Marker building blocks

Building block	Displays
<$startrange>	Beginning of the page range
<$singlepage>	Single page number
<$nopage>	No page number

Note: If the <*$singlepage*> building block isn't used before the "types used in pies" entry, the preceding <*$startrange*> building block is still in effect.

To indicate the end of a range, use the <*$endrange*> building block in the entry's last occurrence.

Also, FrameMaker automatically creates page ranges if you use the <*$autorange*> building block on the reference page containing the Index text flow. Using this method, you don't have to type the <*$startrange*> and <*$endrange*> building blocks in each index marker; however, you can't specify ranges for just a few entries. The <*$autorange*> building block overrides the <*$startrange*> and <*$endrange*> building blocks, so you can't use both. See "Text flows" for more information.

Text flows

In your FrameMaker document, the reference pages contain text flows for generated lists. By using building blocks in these text flows, you can customize the format of your generated lists—you don't have to reformat every time you regenerate the lists.

Table of Contents

You can set up the Table of Contents text flow to automatically format the table of contents with the correct paragraph tags. For example, you may want your table of contents to look like this:

Chapter 1: At Home with Amanda Steubing
Baking apple pie 11
Building gazebos 15
Mending socks 19

Instead of manually applying these tags every time you generate, type the following building blocks in the Table of Contents text flow and assign the appropriate paragraph tags:

`ChapterTOC ¶ tag` —— **Chapter 1:** <$paratext>
`Heading1TOC ¶ tag` ———— <$paratext> . . . . . <$pagenum>

Table 9 on page 8 describes how FrameMaker reads these building blocks.

Table 9: Table of Contents building blocks

Building block	Displays
<$paratext>	Paragraph text
<$pagenum>	Page number

Every time you generate the table of contents, FrameMaker reads the Table of Contents text flow and formats the table of contents accordingly.

Index

You can also set up an Index text flow on the reference pages. This flow operates like the Table of Contents flow, but it uses different building blocks. As mentioned earlier, you can automatically create page ranges by using the <*$autorange*> building block in the Index text flow. To format your index like the one on page 6, include the following building blocks in the Index text flow:

```
Level1IX ¶ tag ————— Level1IX
Level2IX ¶ tag ————— Level2IX
SeparatorsIX ¶ tag ——— 1, 2–3
SortOrderIX ¶ tag ——— <$symbols><$numerics><$alphabetics>
GroupTitlesIX ¶ tag — Symbols[\ ];Numerics[0];
                      A;B;C;D;E;F;G;H;I;J;K;L;M;N;
                      O;P;Q;R;S;T;U;V;W;X;Y;Z
IndexIX ¶ tag ————— <$autorange><$pagenum>
```

To create index entry autoranges, insert the <*$autorange*> building block before the <*$pagenum*> building block in the IndexIX paragraph. For more information on Index text flows, see the FrameMaker documentation.

HTML

The HTML text flow consists of macros that convert FrameMaker elements to HTML elements. You can include certain building blocks in these macros to display descriptive information on the body page. For example, you can set up conversion macros to insert navigation bars in your document. These headings help the user navigate through linked HTML pages.

Table 10 describes the building blocks you can use in HTML conversion macros.

Table 10: HTML Building Blocks

Building block	Displays
<$defaulttitle>	Text of the first document heading
<$nextsubdoc>	URL of the next document
<$prevsubdoc>	URL of the previous document
<$parentdoc>	URL of the first document
<$variable[*varname*]>	Text of the variable

To customize the HTML conversion macros, display the HTML text flow and find the System Macros table. Add the following information to the table:

Macro Name	Replace With
EndOfSubDoc	<A HREF="<$nextsubdoc>">NEXT

The EndOfSubDoc macro inserts text at the end of every subdocument. In the above example, it will insert the word "NEXT" and a link to the next page. See the FrameMaker documentation for more suggestions on using HTML conversion macros.

Conclusion

Building blocks make FrameMaker a powerful tool. By automatically displaying information, they save you time and help ensure consistent formatting and terminology.

Contacting Us

If you have any questions about Scriptorium Publishing Services, Inc., contact:

Scriptorium Publishing Services, Inc.
PO Box 12761
Research Triangle Park, NC 27709
Phone (919) 481-2701
Email sales@scriptorium.com
Web site http://www.scriptorium.com

