

# Locking down document design

Steve Rickaby explains how to create templates in FrameMaker that contain all the formats required to implement your document design.

This article is written in response to a reader's request for basic instructions on creating a template in FrameMaker. Because this is rather a large subject, it is only possible to scrape the surface here rather than give explicit instructions for all the steps involved. Hopefully, however, the article will provide a sufficient outline from which readers can fill in the mechanical details by referring to the relevant sections of detailed FrameMaker documentation. This article outlines only one way of creating a new template, and does not consider the creation of structured templates; for more on that subject, see the articles on structured FrameMaker published in the Spring 2007 to Spring 2008 issues of *Communicator*.

You can of course develop a template by editing one of the samples supplied with FrameMaker to suit your purposes, but here I will consider the process from scratch.

## What is a FrameMaker template?

It is very important to consider this question at the outset, otherwise anyone less familiar with FrameMaker will be forgiven for confusion. Probably the most important point is that FrameMaker templates work very differently to Word templates. While in Word a document can be linked to a template and have its formats updated automatically from the template (and *vice versa!*), no such linkage exists in FrameMaker. A FrameMaker template is best thought of as an *example document* that can be *instanced* (copied). Once a template has been instanced, updates to the formats it contains are under the control of the user of the instanced document—as are updates to the original template—but the two are not linked.

Both approaches have advantages and disadvantages. The chief disadvantage to the FrameMaker approach is that an instance of a template can 'drift' away from the design of the original template as a result of user modifications to its formats.

A FrameMaker template is—or should be—a complete document or set of documents that contains all the formats required for the text design: document, page, colour, paragraph, character and table, as well as embellishments such as graphics.

## Steps in creating a template

We can break down the process of creating a FrameMaker template into the following broad activities:

- Designing the template
- Implementing the template
- Maintaining the template and its instances.

The sections that follow discuss these tasks in detail. In the article, the terms 'template' and 'document' in all cases also imply a group

of templates and instanced documents, for example the various parts of a manual or book.

## Designing your document

Typographic design is an engaging blend of art and craft. Here we can only address the craft, the rude mechanicals of design, by outlining the steps involved. If you lack expertise in the artistic aspects of design—and few technical communicators are also trained graphical artists—a good way to pick up some relevant skills is to look at good examples of existing work and try to extract and emulate the principles. Key areas are:

- Page size and shape
- Page layout
- Choice of fonts
- Paragraph formats
- Use of graphics
- Use of colour
- Use of structural embellishments such as *callouts*
- Table design.

The aspects that have the greatest impact on the appearance of a document are probably page size, *text page* size—the area of a page occupied by type, that is, the paper size minus the margins—font, line spacing (*leading*) and the use of colour.

A useful approach is to 'design with white space': concentrate on the arrangement of the parts of a page that are empty. It has been said that the art of good typographic design centres on the use of empty space.

In real-life contexts your creative freedom may be curtailed by the needs of specific documentation projects, although I hope less so than on a past project of mine where the client's approach was 'You can do what you like as long as it's A4!'

## Implementing your design

Assuming that you have covered the basic aspects of your document design, as described above, what are the steps needed to implement it? The following is a suggested order of approach, but the only fixed ordering is that the master pages should be created first. In fact, because design features interact in FrameMaker, a certain amount of iteration and successive refinement is inevitable.

I assume that you are creating the template for a book or manual; for simplicity, the discussion that follows omits more detailed issues such as conditional text, cross-reference formats and user variables.

### Creating master pages

A book is likely to have a common chapter design that is repeated throughout, so that is a good place to start. Using the **File > New > Document** command, click on the **Custom** button to display the dialog shown in Figure 1. Here you can set

up the page size, margin dimensions, column count, units for measurement and whether the document is to be single or double sided (in our example, the latter). The figure shows typical settings for a manual or textbook. Note that the **Inside** and **Outside** margin settings become **Left** and **Right** if single-sided is chosen.

Completing this dialog and clicking **Create** causes FrameMaker to create a blank document with the attributes, including dimensions, that you specify. It displays a blank page. Now display the master pages (**View > Master Pages**), where you will see that FrameMaker has created a **Left** and a **Right** master page, both of which have both header and footer frames. Delete whichever of the header or footer frames you do not need from both master pages if your intended design does not use them.

Paragraph Designer (**Format > Paragraphs > Designer**). Typically, a minimum set of paragraph formats you might need would include:

- Two or more heading formats
- A body paragraph format
- A bullet list format
- A numeric list format
- An indented format for list run-on paragraphs
- A figure caption format
- Cell heading and cell body formats.

If you display the paragraph catalogue, you will see that FrameMaker has created most of these for you, although you will need to use the Paragraph Designer to modify the formats to suit your design. Here are some useful tips when working with paragraph formats:

- Name formats to describe their purpose rather than their appearance.
- Try to keep the number of formats to a minimum.
- Don't forget to enable the 'keep with next' property for heading formats, so that headings cannot fall at the end of a page.
- If you find yourself creating paragraph formats to implement special functions that are not intended to be selected by authors, such as header/footer formats, prefix their names with 'z\_' or '~' to group them at the bottom of the paragraph palette.

It is helpful to copy some sample text into your template document temporarily so that you can see the effect of your changes to paragraph formats.

You can now return to your master pages and create any running headers or footers that your design needs. Display the master pages, click in the header or footer text frame and use the **Special > Variable** command to display the Variable dialog. Select the **Running H/F** variable to be used, click on **Edit Definition**, and define the variable to pick up the required paragraph tags. For example, using **Running H/F 2** to echo the text of the current **HeadingA** tag would require a definition of the form:

```
<$paratext[HeadingA]>
```

Repeat this process for the other headers or footers you need. Use the paragraph format of the header or footer to format and tab the header/footer fields, and add page numbers using the **Current Page #** variable. Remember to make your page layouts mirror images for double-sided documents, for example always placing the page number towards the outer edge of the page.

Figure 2 shows the typical appearance of a pair of page footers for a textbook. Here **Running H/F 6** and **Running H/F 1** are used to echo the chapter number and title respectively, while **Running H/F 2**

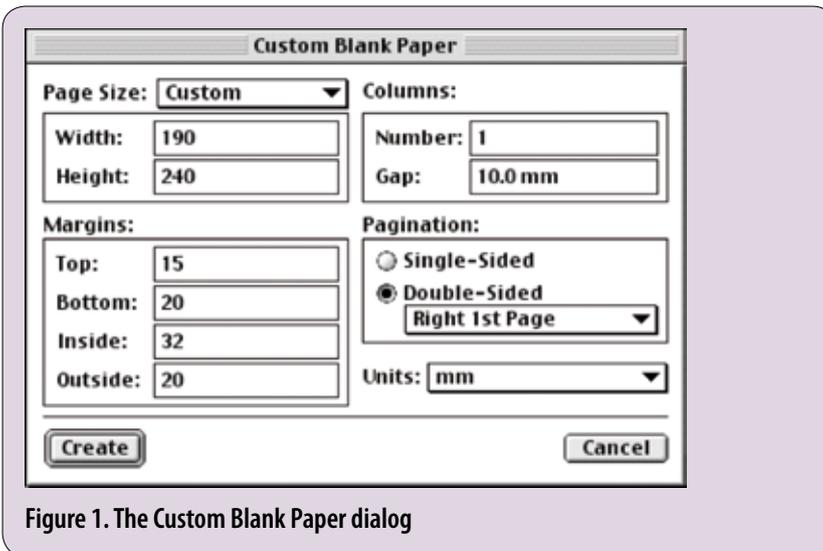


Figure 1. The Custom Blank Paper dialog

You would normally add the constructors for running headers or footers at this point, but here is where iteration is necessary, because header and footer constructors reference paragraph formats and you have not yet created any. You may also wish to create a different master page for the first page of each chapter, for example with space left for a chapter heading, which you can do with the **Special > Add Master Page** command. (This command is only accessible when master pages are displayed.)

There is not space to detail the use of colour in this article, but this is the point at which it would be convenient to define any colours your design uses (**View > Color > Definitions**).

#### Creating paragraph formats

You create paragraph formats using the

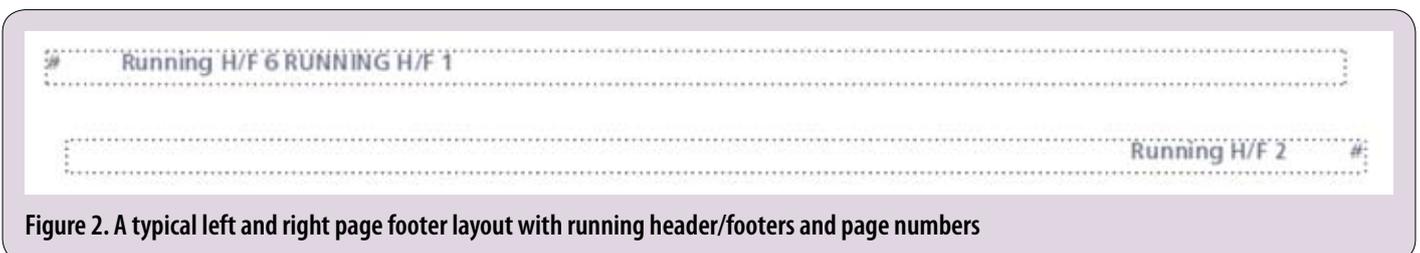


Figure 2. A typical left and right page footer layout with running header/footers and page numbers

echoes the current major (A) heading.

**Running H/F 1**, the chapter title, has been set to display in all-caps using a character format.

#### Creating character formats

Character formats are used to apply special markup to specific words, phrases or sentences, and are also used to format specific parts of cross-references, paragraph formats, header/footers and so on.

Creating the character formats you need in your document design follows a similar course to that for paragraph formats, but using the Character Designer (**Format > Characters > Designer**) instead. It is important to remember that in FrameMaker a character format specifies *modifications* to the underlying font by applying only those character attributes you define in the format. So, for example, if you want to define a format to set a word in italics for emphasis, you must select **Commands > Set Window to As Is** using the pop-up in the Character Designer before selecting **Italic** from the **Angle** field. This ensures, for example, that this format will always use the font, size and colour of the underlying text, as these attributes are not specified in the character format. It will, therefore, work as well in headings, say, as in body text.

We have omitted another iteration here for simplicity: you may need to reference one or more character formats in your paragraph formats, in the **Character Format** field of the **Numbering** pane of the Paragraph Designer. A simple example would be if you wished to use a font or colour other than the body font for the numbering character in a list format. To do this, you would create a character format that specified, say, the Zapf Dingbats or Wingdings font for a bullet character, then reference that format in the paragraph format as described above.

#### Creating table formats

In FrameMaker a table format specifies the appearance of a table (whether it has a title, the ruling and shading styles, the space around the cell contents, the space above and below the table). It does *not* specify the number or width of columns, the number of rows or the presence of heading rows: these are set when a table format is instanced in a document.

A new FrameMaker document is created with two default table formats, **Format A** and **Format B**. Neither is much use other than as a basis for modification. For this you use the Table Designer (**Table > Table Designer**), which has panes for spacings and alignment, default column ruling and default column shading. A simple procedure to create a new table would be as follows:

1. Insert an instance of one of the default table formats in your template document. FrameMaker displays the Insert Table dialog (Figure 3).
2. In this case the number of columns and body rows does not matter, but you should include a heading or footing row if you wish these to be part of your table format, as FrameMaker stores the paragraph format used for these as part of the table format.

3. Using the Table Designer, modify the appearance of the table to suit your design. You might, for example, remove vertical inter-column, left and right rules and shade alternate rows for a 'contemporary' appearance (Figure 4).
4. If you wish your table format to use paragraph formats for the heading/footing rows and cells that differ from the defaults, **CellHeading** and **CellBody**, change the paragraph formats in the relevant cells in the table instance.
5. If your table format has a title, you can change the format for this (which defaults to **TableTitle**) using the Paragraph Designer.
6. Finally, save your table format by entering a new name in the **Table Tag** field of the Table Designer and clicking **Apply**.

At this point you have completed a significant part of the construction of a book template, so, as well as saving your chapter template, it helps to create a book file to contain it (**File > New > Book**). You can then design any other templates for non-generated files you need, such as a title page, add them to the book and import your formats into them from the chapter template using **File > Import > Formats**.

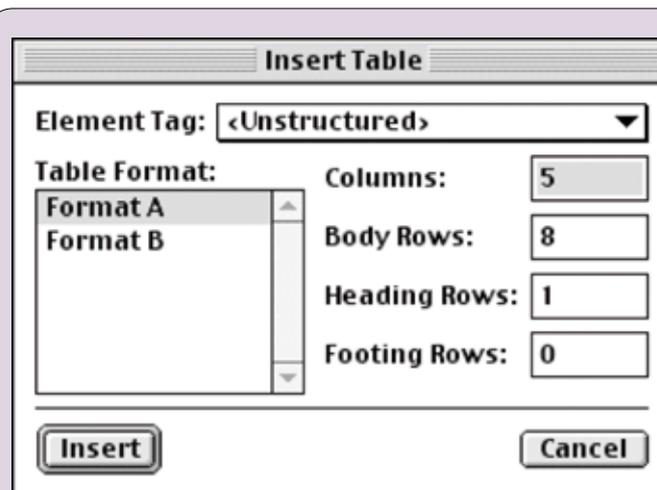


Figure 3. The Insert Table dialog

Column heading	Column heading	Column heading
Row contents		
	Row contents	
		Row contents
Row contents		
	Row contents	

Figure 4. A simple contemporary table design

#### Creating table of contents, index

Within a FrameMaker book, tables of contents, indexes and other files such as lists of figures are *generated* files, added using the **Add** menu from the book window. That is, FrameMaker provides commands to regenerate such files based on the contents of your actual book when the book

is updated (**Edit > Update Book**). How then can templates for these book parts be created and used?

The answer is to generate and format table of contents and/or index files for the book template (you will need to add at least a few headings and index markers in your template files), then format those files in the same way as you have done for your other template files. Then, unless you choose to use the template by instancing (copying) the entire template book—a valid approach—you apply the formatting of the corresponding generated files in your target book as follows:

1. Specify and generate the table of contents/index in your target book. FrameMaker will use its defaults to create files that have valid contents but do not correspond to your template design.
  2. Open the generated file in the template.
  3. Import (**File > Import > Formats**) all the attributes from the book template's generated file.
  4. Regenerate the relevant file in the target book.
- The last step is necessary because FrameMaker only applies the formatting entries on the reference pages of a generated file when the file is generated, not when you import the reference pages.

The exact formatting of paragraphs in generated files is controlled by these formatting entries; this was covered in *Lists: A field of combat?* (*Communicator*, Winter 2005). They are located in special flows on the reference pages (**View > Reference Pages**)—TOC for a table of contents and IX for an index. For example, including a paragraph format **HeadingA** in a table of contents results in entries in the table of contents that have a paragraph format of **HeadingATOC** and a default formatting entry in the TOC flow of the table of content's reference pages of:

`<$paratext> <$pagenum>`

You can modify this to include tabs, heading number and so on, but that is beyond the scope of this article.

#### *Refining the design*

As you will have realised by now, the design components in a FrameMaker document interact: paragraph formats can reference character formats, table formats can reference paragraph formats, headers and footers can use both paragraph and character formats, and so on. If you are working in colour, which there has not been space to detail here, some or all of these may also require colour definitions. Thus some iteration around the procedures described here is normal.

To get a realistic view of your design, it is useful to work with a block of dummy content that includes instances of all your page, paragraph, character and table formats. There are several free programs that will generate dummy text like the traditional 'Lorem ipsum': online ones at [www.lipsum.com](http://www.lipsum.com) and [www.blindtextgenerator.com](http://www.blindtextgenerator.com), and stand-alone ones at [webscripts.softpedia.com](http://webscripts.softpedia.com) (search for 'lorem') and [www.fika.org/davew/judebear](http://www.fika.org/davew/judebear) for Macintosh.

As you refine your page, paragraph, character and table formats, don't forget to import those that are relevant into other files in the book template (**File > Import > Formats**). However, be very careful here in selecting what you want to import,

as uncontrolled import of, say, page layouts can overwrite a lot of your hard work! If you discover that you have done this by accident, close the relevant file without saving and then re-open it and repeat the format import with the correct options.

This begs the question of how much commonality you should aim for across a multi-file template. Clearly a title page template will have a different set of master pages to, say, a chapter template, but what about paragraph, character and table formats? To a certain extent this is a matter of taste and expediency; one option for a book is to use a chapter template as a repository of all formats, and include only those that a specific book part actually uses in other book parts such as title, table of contents and index templates.

With suitable design of numbering, such as using the `<$chapnum>` variable and setting it to alphabetic from the book level (**Format > Document > Numbering**), a chapter template can also serve as an appendix template.

#### **Maintaining a template**

This activity really falls into two parts, one of which is a lot simpler than the other:

- Keeping a 'clean' reference copy of the template in a safe place.
- Dealing with the tendency for instances of the template to 'drift' away from the initial design, a phenomenon that Sarah O'Keefe refers to as *tag creep*.

The first of these is more or less self-evident; in a documentation department or team, it is common for one member of staff to take on this role (although perhaps not to the extent as in one Interleaf team in which I worked where the 'administrator' flatly refused to tell anyone else how to change anything).

Dealing with tag creep is a more complex issue. It is hard to create a complete and perfect template at the outset of a documentation project, and equally it is natural for authors, if at all familiar with FrameMaker, to 'knock up' a homebrew paragraph format to deal with some layout requirement not allowed for in the template. In structured FrameMaker this can be made impossible, but not so in unstructured FrameMaker.

By reimporting the required formats from the template (**File > Import > Formats**) with the **Manual Page Breaks** and **Other Layout/Format Overrides** checkboxes selected, you can go a long way towards forcing the appearance of a document back to the appearance dictated by the template. However, there are dangers in this:

- If an author has used *local formatting*—formatting applied without a tag—this will be lost.
  - As FrameMaker's format import is *additive*, any formats the author has created that are not in the template will remain in the target document.
- Dealing with such issues could be the subject of another entire article: the plug-ins offered by Silicon Prairie Software ([www.siliconprairiesoftware.com](http://www.siliconprairiesoftware.com)) offer more advanced tools to help manage paragraph, character, table formats, master pages and so on at extremely reasonable cost. I recommend them. **C**

#### **Steve Rickaby BSc MISTC**

has been a freelance technical author and editor for 17 years and has used FrameMaker for most of that time.

E: [srickaby@wordmongers.com](mailto:srickaby@wordmongers.com)

W: [www.wordmongers.com](http://www.wordmongers.com)