Mastering master pages: part 2 of 2

FrameMaker's master pages allow you to do more than just control the layout of your document's pages. Steve Rickaby looks at further ideas.

The Summer 2006 article in this series covered the basics of master pages in FrameMaker. Now we can look at some of their more interesting and unusual applications.

Conditional objects on master pages

Jane Dards' article *All change!* in the Spring 2005 issue of *Communicator* introduced conditional text. Briefly, conditions in FrameMaker are represented by condition tags, which exist independently of paragraph and character styles, and are applied to text in a similar way to character tags. For those more familiar with Microsoft Word, conditional text is a little like Word's hidden text, except that you can define as many condition tags as you need in FrameMaker.

The power of conditional text comes from the ability to switch it on and off — make it visible or invisible — at the document and book level. For example, two variants of the same document can be created with two condition tags, *A* and *B*. The two document versions could then be represented as:

- Visible = (untagged text + tag *A*), and
- Visible = (untagged text + tag *B*).

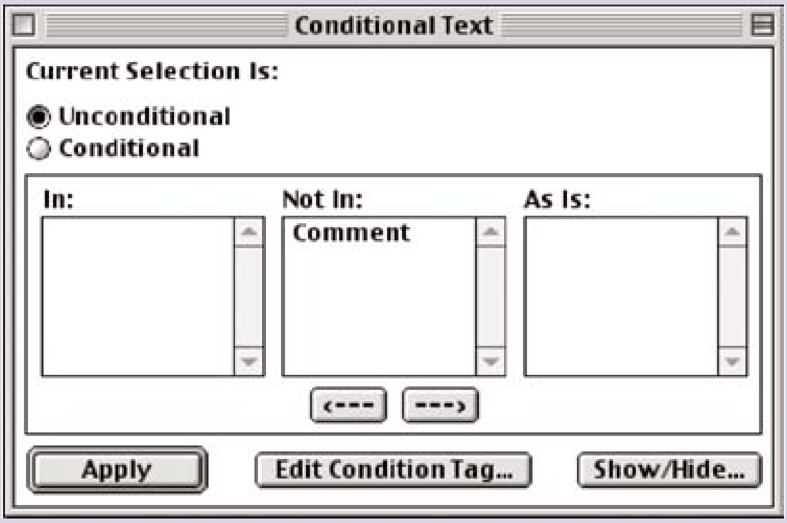

Figure 1. The Conditional Text dialog with the default Comment tag

How can we use conditional text with master pages? In the simplest case, text and graphics on master pages can be made visible or invisible under the control of condition tags in the same way as text and graphics within a text flow. This means that you can vary the appearance of body pages without resorting to changing master pages or reapplying them. You can try this for yourself:

- Create a blank single-sided document in FrameMaker by selecting **File>New>Document**… and clicking on **Portrait** under **Use Blank Paper**.
- Display master pages by selecting **View>Master Pages**…
- Place a heading on the master page by entering text in the header frame.
- Select the heading text, then select **Special>Conditional Text**…
- In the Conditional Text dialog (Figure 1), double-click on **Comment** to move it to the **In** column, then on **Apply**.
- Redisplay body pages with **View>Body Pages**.
- Click on **Show/Hide** in the Conditional Text dialog.
- Use the Show/Hide Conditional Text dialog to make your heading disappear by clicking on **Show** and double-clicking on **Comment** to move it to the **Hide** column, then clicking on **Set**.
- Reverse the process by clicking on **Show/ Hide**, then on **Show All**, and finally on **Set**.

This very simple example shows how you can vary the behaviour of a single master page using condition tags, allowing you to create multiple versions of a document that share the same master page or set of pages.

Sadly you cannot vary layouts easily, as FrameMaker does not allow you to conditionalise text frames on a master page, not even background text frames. You can however conditionalise anchored frames. These can be anchored in a page header, page footer or other background frame on a master page, allowing you to switch graphics on and off with condition tags.

To create margin graphics that can be controlled in this way, add a new background text frame in the margin of your master page and anchor your graphics frame there. Figure 2 shows an example, where two condition tags are used to alternate text and graphics.

The properties of anchored frames can allow us to do much more than this, however, as we will see in the next section.

Dynamic objects on master pages

With a little cunning, it's possible to place *dynamic* objects on master pages. By 'dynamic', I mean objects on a master page that can be made to move on body pages that use the master page, under the control of text features such as autonumber counters, paragraph formats, markers, and so on. This allows the same master page to display differently depending on its context.

Why might one want to do this? Here are a couple of examples:

- Graphical indicators that show a reader's progress through a procedure, a training exercise, and so on. Figure 3 shows a simple example, in which the step labels are static, but

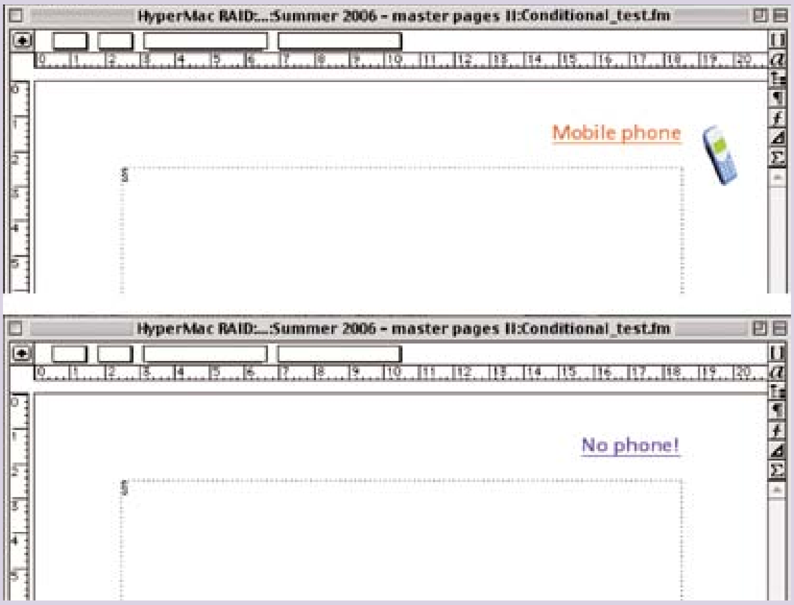
HyperMac RAID:...:Summer 2006 - master pages II:Conditional_test.fm
Mobile phone
HyperMac RAID:...:Summer 2006 - master pages II:Conditional_test.fm
No phone!

Figure 2. Conditional text and graphics on a master page

the arrow moves. Only one master page is used.

- *Fore-edge tabs*—tabs that bleed off the page to mark the chapters of a book when closed and which increment automatically for each chapter.

The fore-edge is the side of a book opposite its binding. It has an interesting history: in the seventeenth and eighteenth centuries, when gilding was fashionable for books, affluent book buyers would sometimes commission water-colour paintings on the fore-edges of their books that could only be seen if the fore-edge was splayed at an angle. Such books are now rare and command high prices. Our purposes are a little more prosaic: in modern textbooks, fore-edge tabs help readers to find a given chapter or section quickly.

But how can an object on a master page be made to move? The answer is to take advantage of the properties of anchored frames. If you place an anchored frame containing a graphic into text using the **At insertion point** option, the frame moves with the text as more text is added before the frame's anchor. Exactly the same is true of anchored frames placed in background text frames on master pages. The trick is to relate the 'text' that moves the frame to something on the body pages.

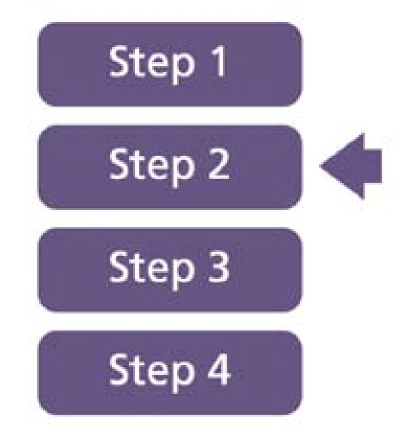

Figure 3. A simple 'animated' graphic

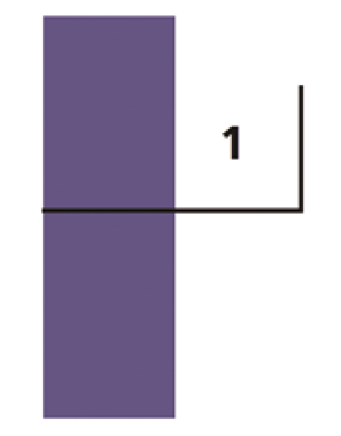

Figure 4. A composite fore-edge tab

There are several ways of doing this. For example, the **Running H/F 3** and **4** variables are mapped by default to the contents of two default marker types, **Header/Footer $1** and **$2**. A method that takes advantage of this is as follows:

- Working in a background text frame on a master page, set an anchored frame in a paragraph with a tag that has suitable tab stop settings. Let's call the paragraph's tag **~MoveGraphic**. (The tilde character keeps the tag's name at the bottom of the paragraph tag palette, as a crude clue that it is not intended for manual selection.)
- Insert the **Running H/F 3** system variable 'in front' of the anchored frame's anchor using **Special>Variable**…
- Import your chosen graphic into the anchored frame.
- Rotate the master page text frame clockwise by 90 degrees if you want the graphic to move down the page.
- Insert **Header/Footer $1** markers on body pages as required using **Special>Marker**…

The graphic in the anchored frame then moves under the control of the number of tab characters (**t**) placed in the **Header/Footer $1** markers. This is how the arrow in Figure 3 is moved.

You can rotate and size the master page's text frame so that the resulting graphic appears where you want on body pages. Varying the font size of the **~MoveGraphic** paragraph tag that formats the **Running H/F 3** variable allows fine control over how far the anchored frame and the graphic it contains move for each successive tab stop in the **Running H/F 3** variable.

So far, so good—but what if you want to move a graphic under the control of some numerical property of a document or book, such as a chapter number? I was faced with a similar problem, and tried many ideas (none of which worked) until I discovered the TabSpace font. This odd font consists entirely of spaces, but the width of each space 'character' is related to its ASCII value. Thus the character that maps to '0' is one space wide, '1' is two spaces, '2 three spaces, and so on. (The actual spaces are somewhere between a thin space and an en-space).

By applying the TabSpace font in an appropriately large font size to the **~MoveGraphic** paragraph tag from the previous example (a couple of hundred points if a fore-edge tab must traverse a page from top to bottom), and preceding the anchored frame's anchor with a **Running H/F** variable defined as **<$paranum[AnotherPara]>**, the movement of the master page graphic is now linked to the autonumber of paragraphs on body pages whose tag is **AnotherPara**. This could be whatever you need: chapter number, section number, or a concealed number (an autonumbered paragraph that does not display its number and exists only to increment the autonumber).

Setting up fore-edge tabs successfully for mirrored left/right pages requires quite a lot of fiddling, but this is the essence of the idea. I have used this method to create decorative fore-edge tabs that move automatically but which can be reset to the top of the page as required using paragraph tags. FrameMaker's support for fractional font point sizes allows very precise control of the 'gearing' of the anchored frame and graphic's movement for each autonumber increment when this method is used.

A couple of hints when manipulating rotated objects on master pages:

- FrameMaker is not very good at allowing you to select and edit rotated objects, to put it mildly. A good way around this is to keep an unrotated copy of your graphics and frames on a reference page or in a separate document, make any edits you need there, copy them to the required master page, and rotate and position them last of all.
- As you increase the font size of the paragraph in which an 'At insertion point' anchored frame lies, FrameMaker changes the **Distance above Baseline** parameter of the anchored frame to compensate for the font size change. If the frame is rotated 90 degrees, this has the effect of moving the graphic towards or away from the page edge. You will need to fine-tune this parameter last of all when positioning fore-edge tabs, particularly if allowing for an off-page bleed. Typical values for fore-edge tabs on a 189 x 246 mm page are 280 points for the TabSpace paragraph tag's font size and 145 points for the distance above baseline.

Composite objects

The example of Figure 3 uses only a single moving graphic, the arrow. When objects such as fore-edge tabs are to be moved, they are likely to contain both text and graphics, as the example of a (left-page) tab in Figure 4 shows.

There are some useful techniques for controlling the placement of text, graphics and so on inside anchored frames. Clearly, the components of the composite object you want to move should be grouped within the anchored frame that does the actual moving. Let's assume that you want to include both graphics and text, and that the text changes, for example with a chapter number, as in Figure 4. Here's how this object is put together:

- The block of colour is a graphics frame with a solid fill. It's dimensioned to provide an off-page bleed to allow for page cropping.
- The ruled half-frame consists of a pair of graphics lines grouped with the coloured frame.
- A transparent text frame is set behind the graphical objects, which is there to hold…
- A one-cell one-row table, anchored within the text frame, which holds…
- The <**$chapnum**> variable, set in a paragraph tag that has its alignment property set to **Center**.

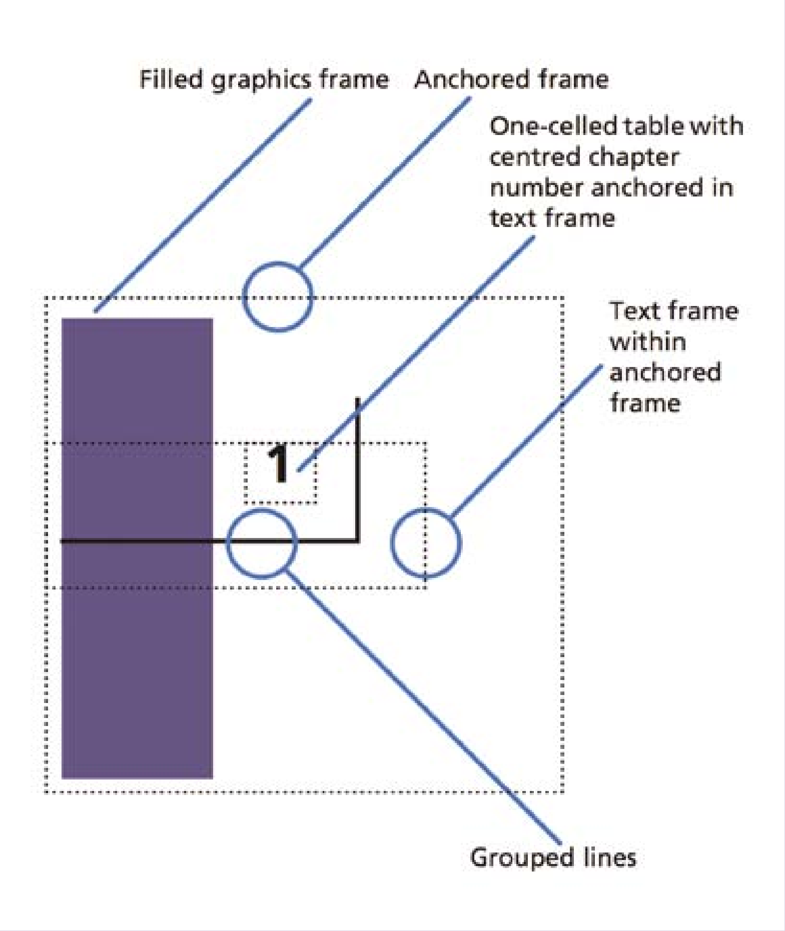
Filled graphics frame
Anchored frame
One-celled table with centred chapter number anchored in text frame
Text frame within anchored frame
1
Grouped lines

Figure 5. The components of a fore-edge tab

Figure 5 details the various parts of the fore-edge tab. The easiest way to 'build' objects like this is in stages, unrotated, grouping them, rotating them and fine-tuning their position only as a final operation.

For more information

Most of the FM features described in this article are covered in the *FrameMaker User Guide.*

I would love to be able to lay claim to the idea for dynamic objects on master pages, but sadly I cannot: I'm indebted to Brad Anderson of Beyond Print for the moving frame idea and other details, and Don Pratt of Telcordia for introducing me to the TabSpace font. This was placed in the public domain by ATIS, The Alliance for Telecommunications Industry Solutions (www.atis.org).

I can supply demonstration files for the animated master page graphics examples if required.

Sarah O'Keefe and Sheila Loring's book *Publishing Fundamentals: FrameMaker 7* contains a method for fore-edge tabs, and is an indispensable reference. If you are interested in fore-edge painting, there are some good examples at the following URLs:

- http://epe.lac-bac.gc.ca/100/202/301/nlnews/nlnews-h/1996/10-11/2810e-07.htm
- www.ibooknet.co.uk/archive/news_jan05.htm#Feature C

Steve Rickaby BSc MISTC has been a freelance technical author and editor for 16 years, and has used FrameMaker for most of that time.
E: srickaby@wordmongers.com
W: www.wordmongers.com