

Composite documents in FrameMaker

FrameMaker enables you to build documents and books from multiple components. Steve Rickaby describes the options.

This article describes techniques you can use to construct *composite* and *variant* documents in FrameMaker, and gives examples of their use. For our purposes, composite documents are documents built from other documents, while variant documents are composite documents whose appearance can be changed using FrameMaker controls such as conditional text.

Most FrameMaker users will be familiar with the idea of a FrameMaker book, in which the book file acts as a container for individual documents, while conferring special properties such as the propagation of numbering threads, the creation of tables of contents, indexes and lists of paragraphs. However, an individual FrameMaker file can also act as a *container document*, by importing other text files into it. FrameMaker uses the term *text inset* for this. When text insets are combined with conditional text, documents can be made to change between variants with very little work.

A hierarchy of methods

All the following are possible in FrameMaker:

- Conditional text insets into anchored frames on master pages can be used to switch page backgrounds.
 - Conditional text insets into body pages can create variant documents by selecting among a population of standard inset documents.
 - Unconditional text insets into body pages can create differing documents by selecting among a population of standard inset source files.
 - Multiple book files can create book variants by selecting from a population of standard chapters.
- These techniques can be combined if required.

Text insets in FrameMaker

Just as FrameMaker allows you to import graphics into a document, it also allows you to import text files with the **File > Import > File** command. As with graphic files, the import can be by reference or by copy, and FrameMaker can be set to update the imported file automatically if the source file changes. However, the options FrameMaker offers differ depending on the type of file that is imported.

Importing a plain text file

If the imported file contains unformatted text, FrameMaker displays the dialog shown in Figure 1. As well as allowing the selection of automatic or manual updating, it gives you the choice of three options:

- **Merge lines into paragraphs.** This is useful if the source file contains paragraph-oriented text. An empty paragraph is treated as a paragraph break.
- **Treat each line as a paragraph.** This is useful

if the source file contains line-oriented text, such as computer source code.

- **Convert to table.** This is useful for delimited text, such as CSV format text. FrameMaker displays the Convert to Table dialog when this option is chosen.

The **Encoding** pop-up offers a variety of alternative character encodings, such as ISO Latin, ANSI and ASCII, to cater for text files originating on different platforms.

Text insets created from unformatted text files can be used, for example, to include sections of source code, or tabular output from databases.

Importing a formatted file

FrameMaker can also import a variety of formatted file formats, but that of chief interest here is import of another FrameMaker document. If you select the **File > Import > File** command and choose a FrameMaker document, FrameMaker displays the dialog shown in Figure 2. There are clearly more options for importing text from FrameMaker documents, but the most significant thing about this dialog is that FrameMaker is offering to import, not an entire file, but a specific *flow*.

Apart from the choice between automatic and manual updating, here is a brief description of the controls in this dialog:

- **Body page flow.** This enables you to import named flows selectively from the source document. While frequently in FrameMaker one might use only the default **A** flow, you can create as many flows as you need, as well as giving them more meaningful names.
- **Reference page flow.** This enables you to selectively import text from a named flow on a reference page. The reference page can be in another document, or in the container document itself. This is explained below.
- **Reformat using current document's catalogs.** This allows tags in the container document whose names match corresponding tags in the imported file to override those tags in the container. Tags with names that do not occur in the container document are unaffected.
- **Reformat as plain text.** This removes all formatting from the imported text. The formatting applied to the inset is then the formatting of the paragraph tag at the insertion point. No tags are imported.
- **Retain source's formatting.** This leaves the imported text unaltered. Tags in the imported text are not added to the container document's catalogs.

The ability to import specific text flows selectively from a source document or reference page flow into a container document suggests uses for

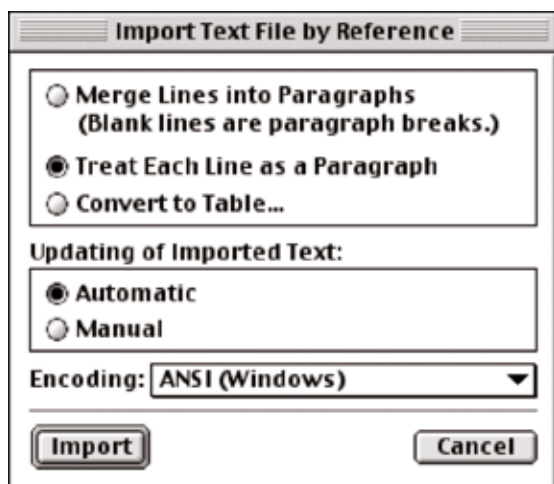


Figure 1. Importing a plain text file

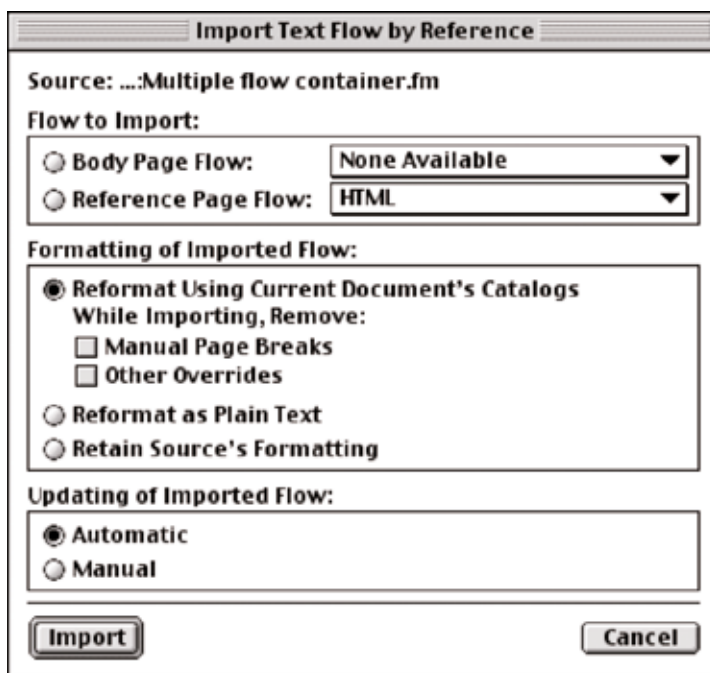


Figure 2. Importing a formatted text file

handling 'boilerplate' text. This is text that is used many times in a document or document set, such as standard warnings, phrases and legal text. These can be placed in their own documents, or on named reference page flows in a FrameMaker template or globally-accessible source document.

If importing named reference flows is more convenient, you can create these in a source document as follows:

1. Select **View > Reference Pages**.
2. Select **Special > Add Reference Page**.
3. Supply a meaningful name for the new reference page.
4. Using the text frame tool, draw a frame on the new reference page.
5. With the new frame still selected, select **Graphics > Object Properties**.
6. Supply a meaningful name for the flow tag in the **Tag** field of the Customize Text Frame dialog.

It is this flow tag name that will appear in the pop-up menu of the Import Text Flow by Reference dialog, Figure 2. (To import a flow from a reference page in the *container* document, select the container document itself in the Import File dialog.)

Conditional page backgrounds

Now that we've covered the basics of text insets in FrameMaker, how can we use them creatively?

When text insets are combined with conditional tags, it is possible to construct documents that can radically change their appearance in response to only small changes in the document's configuration. Suppose, for example, you are working with a document that uses standard background text that has to exist in multiple languages, or a form that needs to exist in several different versions.

We saw in *Mastering Master Pages II* (*Communicator* Autumn 2006) that text frames cannot be conditionalised, not even background frames on reference pages. One way around this is as follows:

1. Place anchored frames in a background text frame on the master page(s) of the container document.
2. Apply suitable conditional text tags to the frames' anchors to allow them to be made visible alternately.
3. Extend the anchored frames to cover the main text flow on the master page. Arrange and align them to suit your application.
4. Import different text insets into each anchored frame.

This arrangement, illustrated in Figure 3, enables you to switch the backgrounds of body pages under the control of conditional text tags, each background originating from a different source document if required.

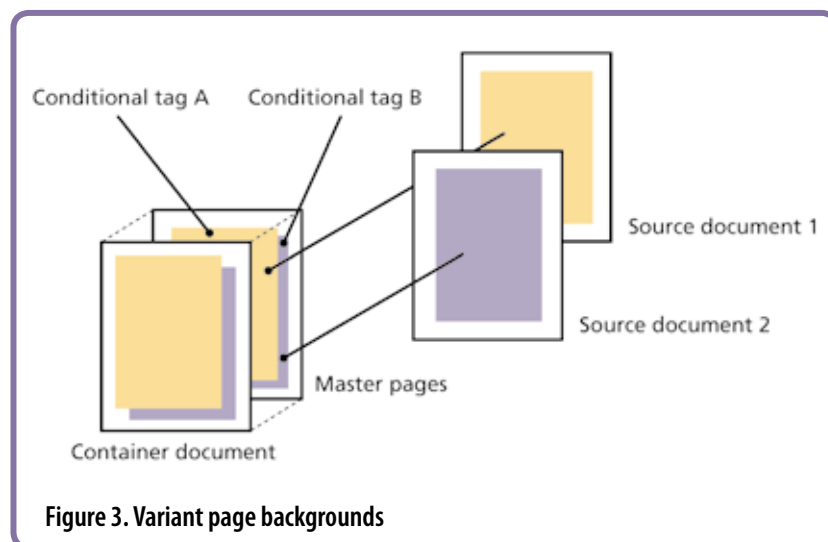


Figure 3. Variant page backgrounds

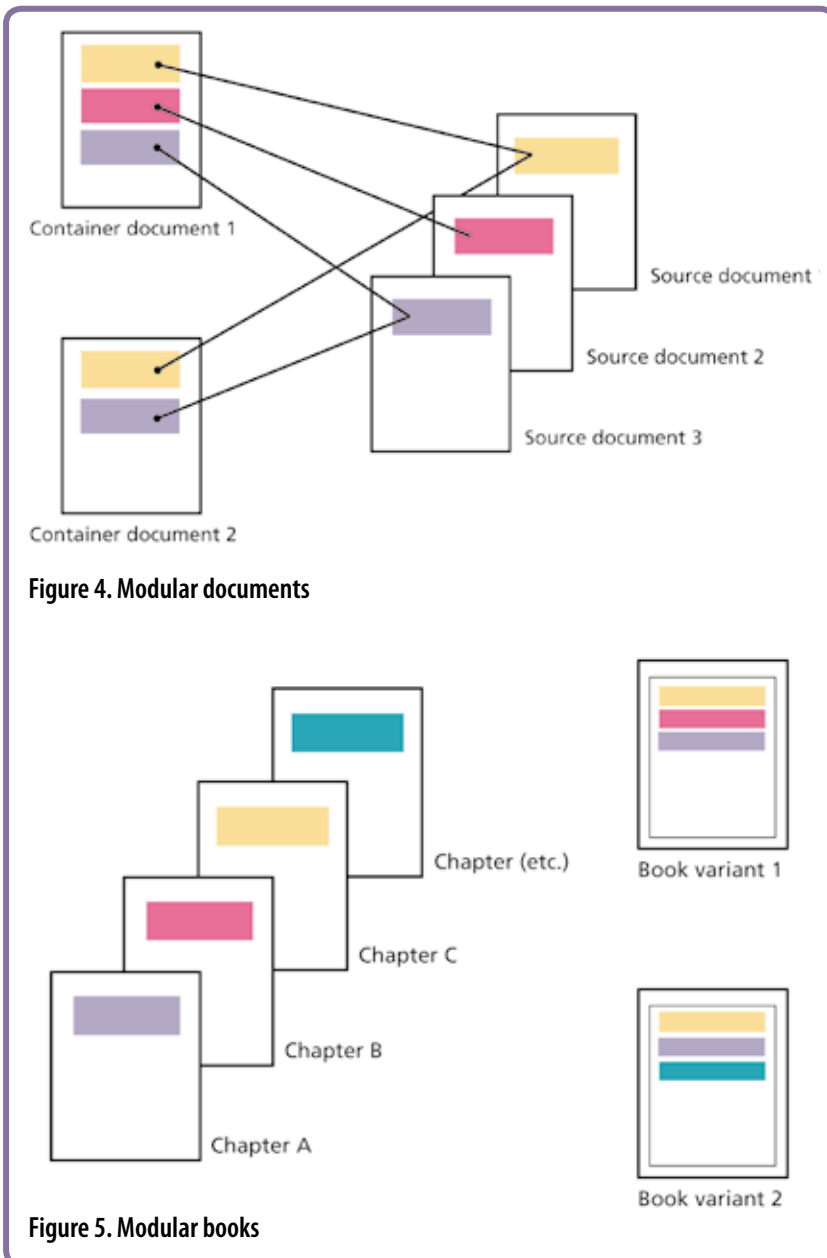


Figure 4. Modular documents

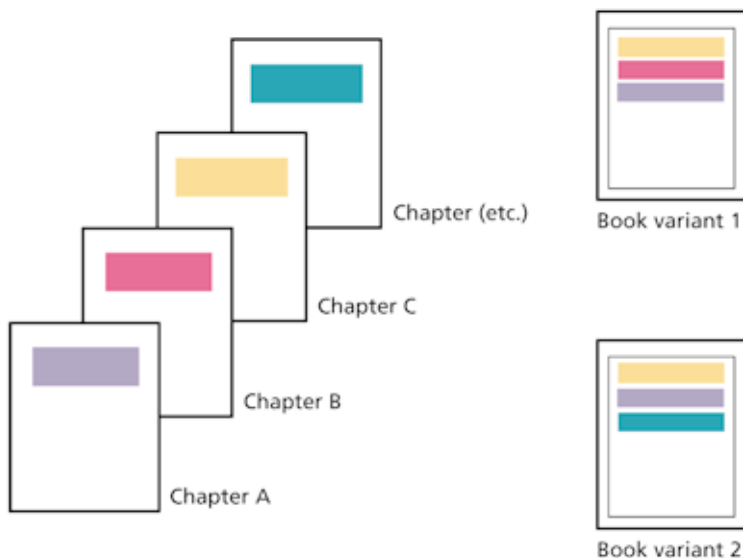


Figure 5. Modular books

Building modular documents

Modular document is the term Sheila O’Keefe and Sarah Loring use in their book *Publishing Fundamentals: FrameMaker 7* to describe documents composed of multiple text insets. This scheme is useful for documentation that is constructed from various combinations of standard texts, such as product documentation that shares some common text.

Figure 4 illustrates the general scheme, in which two different documents are constructed from various combinations of standard texts. In contrast to Figure 3, here the text insets are imported into body pages. Although the figure shows small sub-page text insets for clarity, no such length restriction exists in practice. Equally, the three source documents could be replaced by a single source document in which the standard texts were contained in named flows.

Documents constructed in this way can also make use of conditionalised insets, although you need to take care to avoid excessive complexity and resulting maintenance headaches.

Some snags of text insets

Text insets are not without their downsides. For example, FrameMaker has a habit of dropping conditional tags from text insets when they are updated (a ‘logical bug’?). Placing text insets in conditionalised anchored frames, if your application allows, can avoid this problem. It is also complex to create sustainable cross-references into a text inset: this has to be done in a two-step process, first creating a named cross-reference destination marker in the source file, then creating a cross-reference to it from the container file. Also, insets of FrameMaker documents cannot be placed adjacent to each other in a container document without an intervening blank paragraph, as a FrameMaker bug causes formatting to ‘leak’ from one inset to the next.

The handling of user variables in text insets is complex and can be confusing: in the case of variables that are defined in both the source and the container document, the value of the variable that is used depends on your choice for the formatting of the imported flow — see Figure 2 — which can differ from inset to inset.

Modular book variants


Moving up a level in FrameMaker document organisation, just as for documents in Figure 4, you can construct different versions of books by building them from different combinations of chapter documents. Figure 5 illustrates this arrangement. In the figure the chapters are named alphabetically to illustrate that their ordering, and hence their numbering, depends entirely on the book in which they find themselves. You normally need to update each book before printing to give chapters the required numbering and page sequence.

Ur-books

Text insets imported by reference are excluded from editing, searching and spell-checking in the container document. When working with populations of source documents that you are using to create modular documents, books, or a combination of both, it’s useful to be able to use FrameMaker’s book-level tools for searching and spell-checking the source files. One way in which you can do this is to build a dummy FrameMaker book — ‘dummy’ in the sense that it’s never used to create a printable version — that contains your entire population of source documents.

The term often used to describe this is *ur-book*. There is some debate over the derivation of this term, although ‘ur-’ as a prefix means ‘original’, as in ‘ur-form’. Some prefer to believe that it derives from German, as a truncation of *Überbuch*, which certainly helps to convey the idea well.

An ur-book for the modular documents shown in Figure 4 would contain all three source documents, while an ur-book for the modular book arrangement shown in Figure 5 would contain all the source chapters: their order would be immaterial.

The same technique can be used to build an index of multiple books, by creating a ‘book of books’. 

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