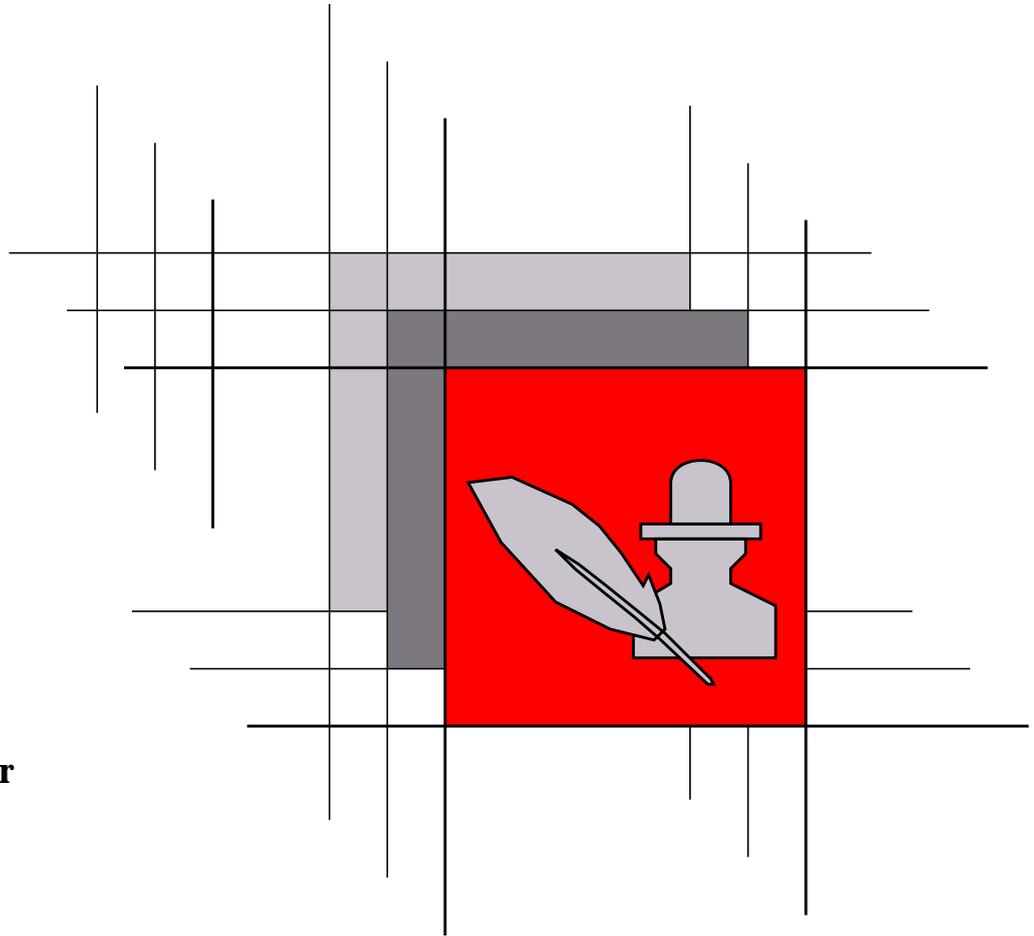




**European Laboratory for Particle Physics
Information, Process and Technology Group
CH-1211 Geneva 23, Switzerland**



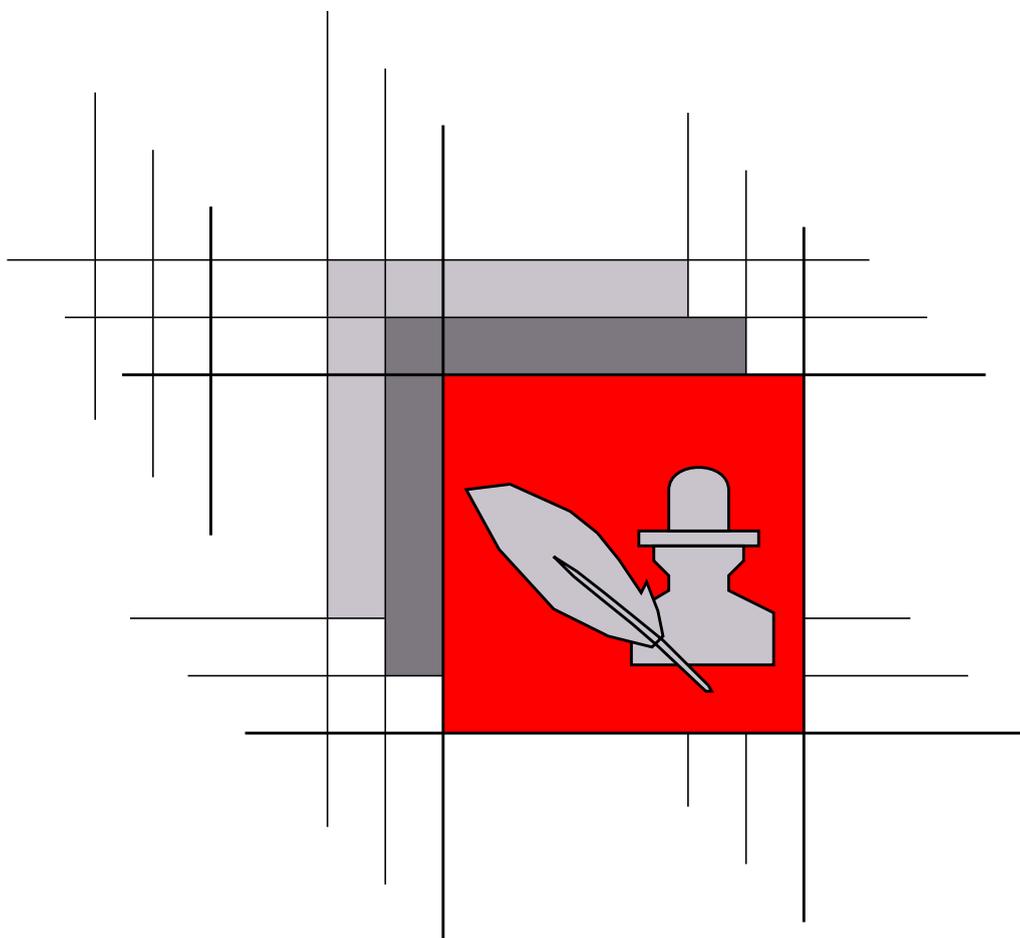
**Mario Ruggier
15 July 1998**

How to use the FrameMaker

User Guide Templates

Release 2.0

<http://www.cern.ch/FrameMaker/>



How to use the FrameMaker

User Guide Templates

Release 2.0

<http://www.cern.ch/FrameMaker/>

Copyright CERN, Geneva 1997 - Copyright and any other appropriate legal protection of this documentation and associated computer program reserved in all countries of the world.

Organisations collaborating with CERN may receive this program and documentation freely and without charge.

CERN undertakes no obligation for the maintenance of this program, nor responsibility for its correctness, and accepts no liability whatsoever resulting from its use.

Program and documentation are provided solely for the use of the organisation to which they are distributed.

This program may not be copied or otherwise distributed without permission. This message must be retained on this and any other authorised copies.

The material cannot be sold. CERN should be given credit in all references.

This document has been prepared with Release 5.5 of the Adobe FrameMaker[®] Technical Publishing System using the User's Guide template prepared by Mario Ruggier of the Information and Programming Techniques Group at CERN. Only widely available fonts have been used, with the principal ones being:

Running text:	Palatino 10.5 pt on 13.5 pt line spacing
Chapter numbers and titles:	AvantGarde DemiBold 36 and 24 pt
Section headings	AvantGarde DemiBold 20 pt
Subsection and subsection headings:	Helvetica Bold 12 and 10 pt
Captions:	Helvetica 9 pt
Listings:	Courier Bold 9 pt

Use of any trademark in this document is not intended in any way to infringe on the rights of the trademark holder.

Copyright CERN, Geneva 1997 - Copyright and any other appropriate legal protection of this documentation and associated computer program reserved in all countries of the world.

Organisations collaborating with CERN may receive this program and documentation freely and without charge.

CERN undertakes no obligation for the maintenance of this program, nor responsibility for its correctness, and accepts no liability whatsoever resulting from its use.

Program and documentation are provided solely for the use of the organisation to which they are distributed.

This program may not be copied or otherwise distributed without permission. This message must be retained on this and any other authorised copies.

The material cannot be sold. CERN should be given credit in all references.

This document has been prepared with Release 5.5 of the Adobe FrameMaker® Technical Publishing System using the User's Guide template prepared by Mario Ruggier of the Information and Programming Techniques Group at CERN. Only widely available fonts have been used, with the principal ones being:

Running text:	Palatino 10.5 pt on 13.5 pt line spacing
Chapter numbers and titles:	AvantGarde DemiBold 36 and 24 pt
Section headings	AvantGarde DemiBold 20 pt
Subsection and subsection headings:	Helvetica Bold 12 and 10 pt
Captions:	Helvetica 9 pt
Listings:	Courier Bold 9 pt

Use of any trademark in this document is not intended in any way to infringe on the rights of the trademark holder.

Why these templates ?

If you wanted to write user documentation, where would you begin from ? What should the document contain ? How should the information be organized ? How should it be presented ? How do you make sure that the document you produce will be usable ?

These templates provide a sorely needed starting point and framework to begin addressing the above issues. They offer an environment in which you can plan, organize and develop your user documentation such that it is coherent, both in content and in presentation.

About user documentation.	vi
The problem addressed by these templates	vii
About this manual	viii
About the templates package	ix
Acknowledgements.	x

About user documentation

“Documentation is a vital but often overlooked part of any technical project. Even when a considerable amount of effort is put into producing and maintaining documentation it may still fail to be useful if the style and layout are inappropriate to the audience and intended purpose.” [3]

Part of the software

User documentation is not an after-thought of the software development process. This archaic approach is a breeding ground of many problems. There may be very little time left to produce the documentation before the final system delivery deadline. Or, due to a lack of documentation, the system may not be reviewable by users prior to its completion, and any change then will be quite expensive.

The only way to give documentation the importance it deserves is to accept the fact that it is an integral part of the system. This *implies* that it is planned for along with the rest of the project, and that appropriate resources are allocated for it.

Advantages of good documentation

A user confronted with a possibly good, but badly documented, system will very likely draw judgement on the system based solely on the documentation. The evaluation will probably stop not much after the initial frustration with the system, due to its inadequate documentation.

Good documentation saves time and money as users can become familiar, understand and properly use the system faster. It can actually help reduce your manpower problem!

Importance of presentation

How easily you can find and absorb what you are looking for in the user documentation of a system, irrespective of whether you are a novice or an expert user of the system, is largely a function of how the document is organized and presented. Manuals should be clearly and consistently laid out, with text set in as readable a manner as possible and with good lists of content, index and other appropriate reference aids. These features are essentially part of the presentation and, in the case of these templates, are all obtainable virtually for free.

As an author, using templates frees you from worrying about the finicky and time-consuming aspects of presentation, giving you more time and energy to devote to the contents.

The problem addressed by these templates

We all understand the importance of user documentation. We also know that producing user documentation that is coherent, both in content and in presentation, is difficult—how often are we frustrated when we are learning to use a new system and cannot find the information we need ?

The problem currently

How do we go about writing user documentation today ? We probably start from a template that, at first sight, does approximately what we want, and then we start writing. As we write we discover that we need to consistently create new format definitions, or wrestle with whatever tool we have adopted to achieve what we very clearly want. The situation is even worse if the authoring effort involves more than one author. Inconsistencies can reach a point such that you completely lose the flexibility that the tool offers you to control a logically tagged document.

The result is more often than not user documentation that has not only taken much too long to produce but is also not as usable as we wished and expected it to be. The program to which the documentation belongs is therefore much less usable than it could have been had the user documentation been better conceived.

Templates for better results

These templates address the above problematic scenario by offering a framework that takes care of the presentation aspects for your user documentation, and helps you organise it. Several authors can work on the same document in a controlled manner. Practical guidelines for multiple author editing are provided in this manual. You do not need to spend time fiddling with the formatting of your material and to wrestle with the tool to do it as you want it to be. Aspects of the document layout such as the numbering, pagination of figures, running headers and footers, tables of content, the index, references, etc., have all been carefully thought out to maximise the usability of your information within. Accepted type setting standards and results from research studies on readability of typographic material have been adhered to as much as possible to maximize the readability of documents produced with these user guide templates [9], [10]. You can just concentrate on the actual content.

The template itself will not change the fact that you have to write *your* user documentation *yourself*, but it is hoped that having a functional presentation already taken care of will help you formulate your ideas easier and better.

Using these templates helps you to not only produce your user guide more quickly—thus for less money—but may also make your final results better than they might have been otherwise.

About this manual

This manual is produced with the STOP [2] variant of the user guide templates documented here. Key issues are thus presented as double page sections which include a full page illustration.

Who and what is it for ?

This manual is for all authors and editors of documentation prepared using the FrameMaker User Guide Templates. It is the only available document describing these templates.

Although it contains many tips on the use of FrameMaker, this manual does *not* substitute the FrameMaker documentation [4]. A basic understanding and familiarity with FrameMaker is assumed.

There is also much to be said about writing user documentation. Different styles and techniques exist for different purposes. Again, although many sensible suggestions are offered, this manual is *not* a description of how best to write user documentation—to learn more on this subject please refer to more dedicated publications, some of which are listed in the bibliography, e.g. [2] and [8].

The chapters

Before starting to edit a user guide with these templates you should first read Chapter 1, "Getting started". This presents the major options and decisions that you should make early on. How to organise your document is also explained here.

Chapter 2, "Editing the content" deals only with how to edit text, figures, tables, and content in general.

Routine procedures for managing FrameMaker books are presented in Chapter 3, "Delivering the document". Other non-routine maintenance procedures are collected in Appendix A, "Set-up and maintenance procedures".

The main concepts of the templates, explained in Chapter 1 and Chapter 2, are conveyed using the STOP method, described in Section 1.1. The remaining more procedural material of the document is written in a more conventional structured report style.

Conventions used in this manual

In general the names of *files* and UNIX *utilities* are printed in italic Courier. The sequence of FrameMaker *menu selections* or the names of FrameMaker *dialog windows* are printed in AvantGarde DemiBold Oblique. FrameMaker **keyboard shortcuts** are printed in bold Courier. *Simple emphasis* uses italics.

Steps in step-through procedures are numbered with the number outdented from the text.

About the templates package

Availability of the templates

The distribution kit is available on WWW from:

<http://www.cern.ch/FrameMaker/>

It is available only for users at CERN or at collaborating institutes.

Required software

The templates are prepared in Adobe FrameMaker Release 5. Automatic conversion to WWW is preconfigured for WebMaker 3.0.

We'd like to hear from you

Send us your comments and suggestions about both this document and the templates themselves. If you notice any errors or encounter a problem when using the templates, please inform us.

To communicate with us send mail to docsys@ptsun00.cern.ch.

Acknowledgements

I gratefully acknowledge all who have helped with this project, in particular:

- Alberto Aimar and Paolo Palazzi, colleagues in IPT, for originating the idea, for convincing many of the importance of such templates, and for their continued support throughout the project.
- All members of the IPT group, from the ECP Division at CERN, for support and infrastructure.
- Dante Borillo, Erwin Dénes and Fred Wickens, courageous users of early versions of the templates. Their feedback resulted in innumerable improvements.

Contents

Why these templates ?	v
About user documentation	vi
Part of the software.	vi
Advantages of good documentation	vi
Importance of presentation	vi
The problem addressed by these templates	vii
The problem currently	vii
Templates for better results	vii
About this manual	viii
Who and what is it for ?	viii
The chapters	viii
Conventions used in this manual	viii
About the templates package	ix
Availability of the templates	ix
Required software	ix
We'd like to hear from you	ix
Acknowledgements	x
Chapter 1	
Getting started.	1
1.1 Two document styles to choose from	2
The structured report or normal style	2
The STOP method	2
Why use it ?	2
Choose the style early.	2
1.2 File organization and set-up	4
Organisation of the template and document files.	4
Setting up your user guide document	4
1.3 Managing template customizations	6
What is a template customization ?	6
Managing customizations	6
Naming convention for new formats	6
Customization checklist	6
1.4 Page size options	8
Different page sizes to choose from	8
Changing the page size	8
Select the page height as early as possible	8

Centralize template customizations	8
1.5 Planning your user guide	10
The task-oriented approach	10
Organize the tasks into chapters	10
The table of contents	10
1.6 Adding and deleting chapters	12
Use your customized template	12
Adding content files	12
Re-adding automatically updated files	12
Adding new content files to the book	13
Deleting files from the book	13
1.7 Managing imported files	14
Advantages of importing by reference	14
Advantages of importing by copy	14
 Chapter 2	
Editing the content	17
2.1 The chapter title page and section headings	18
The chapter title page	18
Sections and subsections	18
2.2 Running text	20
Paragraph formats	20
Character formats	20
2.3 Tables	22
Inserting a table	22
Basic pre-defined table formats: <i>2Col</i> , <i>3Col</i> , <i>4Col</i> , <i>5Col</i>	22
2.4 Figures, listings and snippets (Normal template)	24
Inserting a figure	24
Listings and snippets	24
Inserting	24
Listings longer than 1 page	24
2.5 Illustrations (STOP template)	26
Right-page illustrations	26
Formatting illustrations	26
2.6 Controlling indentation and spacing in listings	28
Inserting multiple spaces	28
Importing a listing by copying	28
Importing a listing by reference	28
2.7 Function calls	30
Formatting separate units such as functions	30
Customizing for more specific paragraph tags	30
2.8 The <i>zany</i> paragraph formats	32
Use only few <i>zany</i> paragraph formats	32
Importing <i>zany</i> formats or creating new ones	32
Using <i>zany</i> formats	32
<i>Zany</i> sections	32
2.9 The front matter	34
The cover and inside cover	34
Editing the frontmatter file	34
Roman page numbering	34
2.10 The appendix and the bibliography	36
The appendices	36
The bibliography	36
References to bibliography titles	36
2.11 The index	38

Importance of the index	38
Building the index	38
Chapter 3	
Delivering the document	41
3.1 Routine book procedures	42
Updating the FrameMaker book	42
Printing to standard sized paper	42
Cutting the paper	42
3.2 The running footer and background text	43
Resetting the variables in a single file	43
Propagating to all other files in the book	43
3.3 Converting to WWW	44
Before saving as HTML...	44
After saving as HTML...	44
Saving as PDF	46
Dealing with changes to the template	46
Appendix A	
Set-up and maintenance procedures	47
A.1 Further thoughts on multi-author editing	48
Version control of the entire document	48
Author submissions of material in progress to editor	48
Managing inter-author cross-references	48
A.2 Changing the page layouts for different page sizes	49
Page layouts for the cover, the index, and all other files...	49
Predefined page layouts	49
Changing the page size	50
Preliminary set-up	50
The cover page file	50
The index file	51
The other files	51
Finishing	52
A.3 Deleting and re-adding automatically updated files.	54
Deleting an automatically updated file	54
Re-adding an automatically updated file.	54
A.4 Updating the formats of your document files	56
File types	56
Updating each file	56
A.5 Copying and creating <i>Zany</i> paragraphs	58
Copying <i>Zany</i> paragraph formats	58
Creating your own <i>Zany</i> paragraph formats	59
Creating a new <i>Zany</i> frame	59
Creating a new <i>Zany</i> paragraph	59
A.6 Splitting a chapter into several files	60
Splitting the content	60
Setting the running header in subsequent files.	60
Bibliography	61
Index	63

Chapter 1

Getting started

Each user guide template comes in the form of a FrameMaker book that is pre-configured with automatically updated lists, such as tables of content and index. To get going you simply need to create a document instance and then chapter and section headings, according to your document plan. A few template options are required to be set at this beginning stage, thus you need to know what they consist of and how to select them.

1.1	Two document styles to choose from	2
1.2	File organization and set-up	4
1.3	Managing template customizations	6
1.4	Page size options	8
1.5	Planning your user guide	10
1.6	Adding and deleting chapters	12
1.7	Managing imported files	14

1.1 Two document styles to choose from

There are two user guide templates to choose from, each supporting a different style of writing user documentation. One template supports the structured report style, with which we are all familiar, and the other, the STOP method, is a hybrid of structured writing and preparation of transparencies for presentations.

The structured report or normal style

All the commonly available user documentation basically follows this ubiquitous style of technical writing, in which documents are organized as hierarchies of chapters, sections and subsections, with each section containing running text intermixed as necessary with figures, tables and listings. It can be quite suitable for almost all types of user documentation.

The STOP method*

The STOP method [2] breaks up each chapter into independent double page spreads. Each double page spread should convey a single complete concept and is made up of a textual page and an illustration page. You can think of each 2-page section as a transparency accompanied with comprehensive textual explanation[†].

“The STOP method can be used whenever a concept can be more clearly expressed by both diagrams and text.

You may suspect that it would be impossible to squeeze all of your ‘complex’ information onto separate double page spreads. However, making the effort to do so is one of the best ways of breaking your information into manageable chunks.” [3]

Why use it ?

“The left side of the brain works with linear and sequential logic (such as text), the right cerebral hemisphere deals with spatial and visual concepts (such as pictures). One of the best ways of imparting information to the reader is through a combination of both words and pictures, thus enabling the reader to use both sides of the brain.” [3]

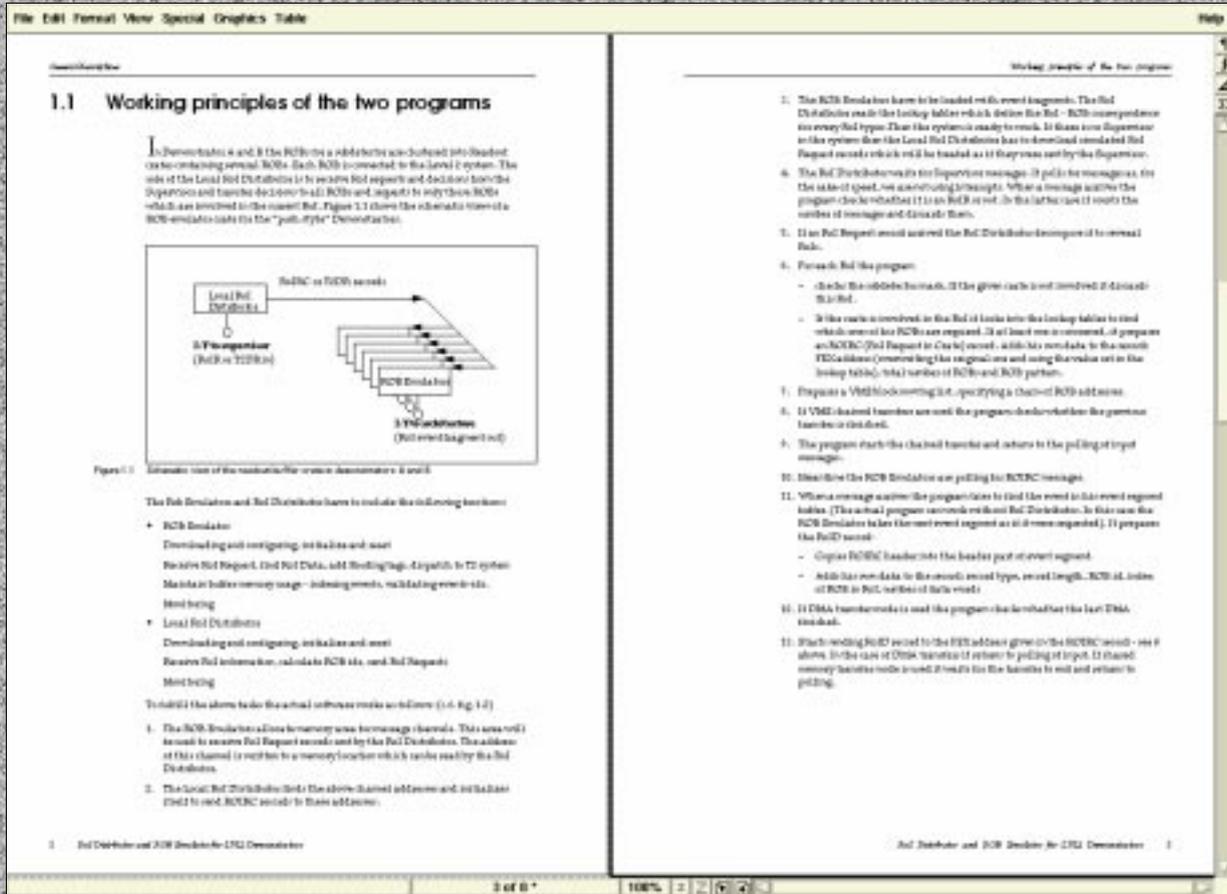
Choose the style early

You should decide whichever of these two styles you adopt for your document as early on as possible as shifting the material from one style to another may require some rewriting and other adjusting.

* The STOP (Sequential Thematic Organization of Proposals) method was invented by Hughes Aircraft Company in the late 1960s [2].

† The STOP template assumes text on the left and illustration on the right. If you would prefer to reverse this, see Section 2.5, "Illustrations (STOP template)".

Normal style



STOP method

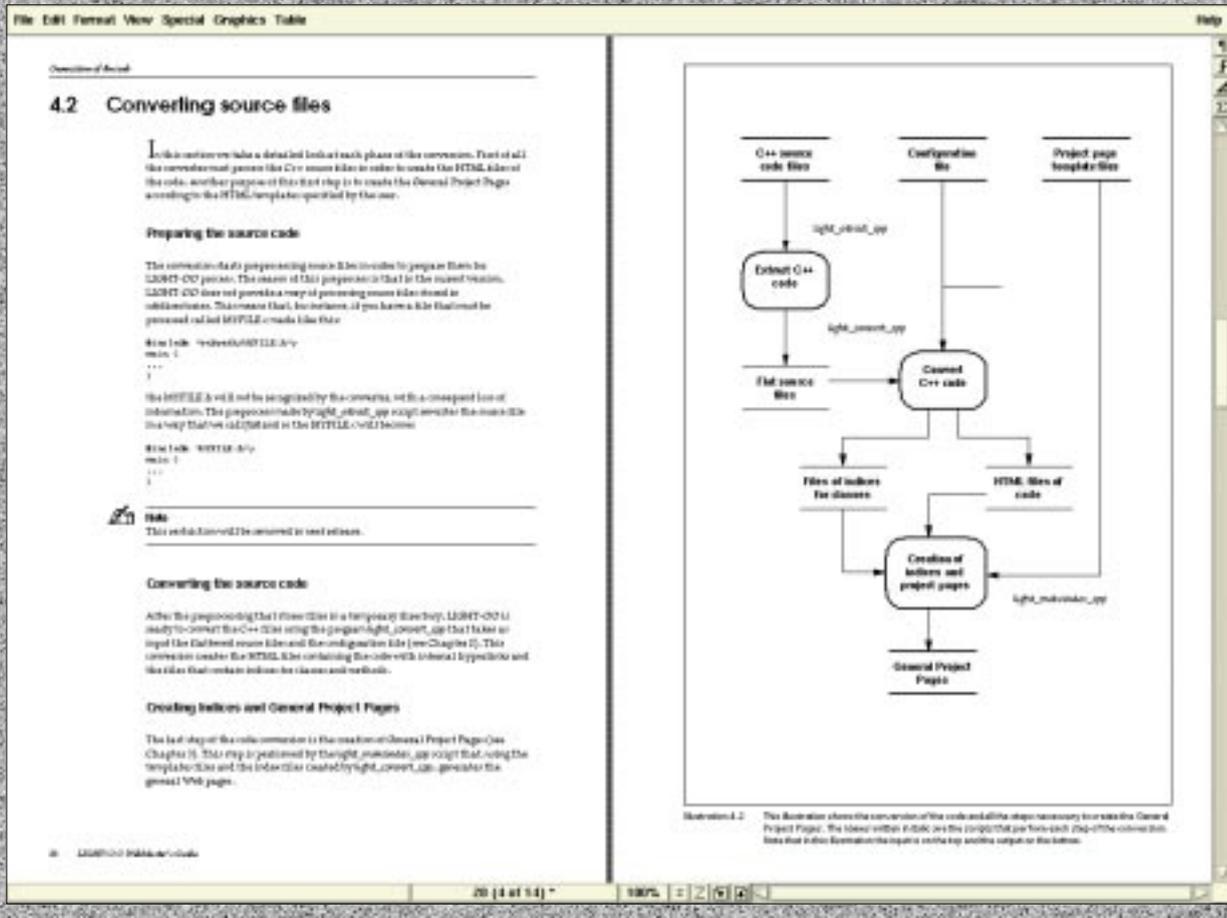


Illustration 1.1 The principal differences between the two document styles can be seen from these examples of 2-page spreads. The normal style does not restrict text to fit only on one page; and figures, tables and listings may be inserted wherever and whenever needed. The STOP example expresses a single concept as a 2-page section, one page for text and one page for an illustration.

1.2 File organization and set-up

The way you organize your files depends on your process, i.e. on how you want to work. For clarity we assume that the normal template is being used, but this discussion applies also to the STOP template.

Organisation of the template and document files

The template is a FrameMaker book file containing several component template files, some of which are editable and others are automatically updated. The template is delivered with the directory structure of the *template* subdirectory in the illustration. Note that individual templates for editable component files, i.e. cover, frontmatter, chapter, appendix and bibliography, are delivered in separate subdirectories.

The document is similarly organized. For multiple authors it is best to have a subdirectory per chapter, as this is easier to manage access rights for, and to re-assemble the book (check-in and check-out of individual chapters). For single authors a flat directory structure may be easier.

See also Appendix A.1, "Further thoughts on multi-author editing" for other related issues.

Setting up your user guide document

1. Make a base dir, such as *userguidebase*.
2. Decide which template model to use, as discussed in Section 1.1, "Two document styles to choose from".
3. Set up the reference template for your document as follows:
 - a. copy the template tar file, e.g. *normal.tar* to *userguidebase*;
 - b. untar and rename the created directory e.g.:

```
tar xvf normal.tar
mv normal template
```
 - c. remove the file *normal.tar*.
4. Customize the reference template for your document as specified in Section 1.3, "Managing template customizations".
5. Set up the directory structure for your document as follows:
 - a. in *userguidebase* create a *document* subdirectory, and in it create the subdirectories according to the way you want to organise your document;
 - b. copy, from the *template* subdirectory, the FrameMaker book file, all the automatically updated files, the cover, the frontmatter and bibliography;
 - c. add blank files for each chapter and appendix as specified in Section 1.6, "Adding and deleting chapters".

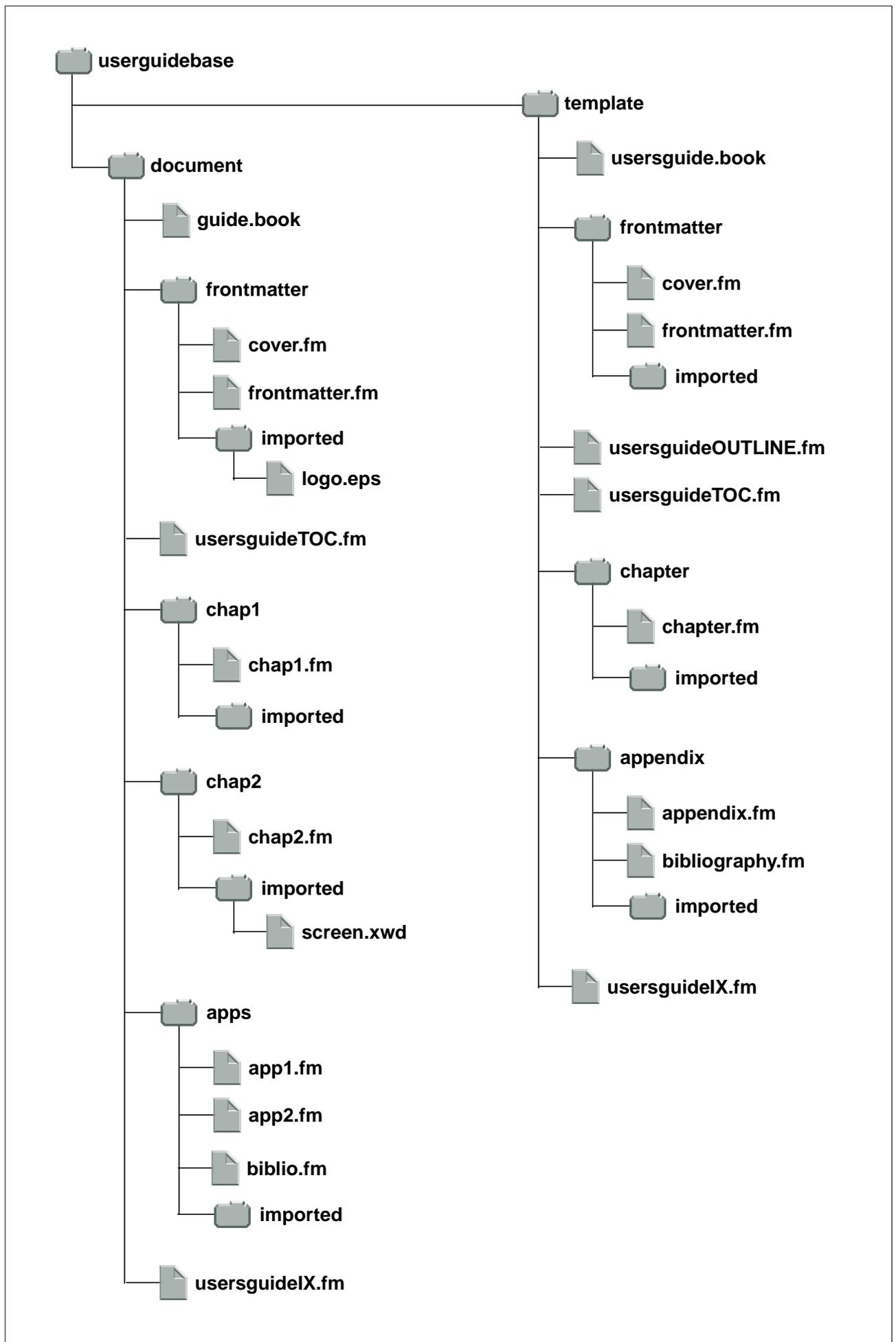


Illustration 1.2 The default file organization of the FrameMaker book template is indicated by the *template* subdirectory. The file structure for the actual document is derived from this and is indicated in the *document* subdirectory.

1.3 Managing template customizations

You may want to add to, or modify, the pre-defined formats in the template. Managing such customizations while still maintaining integrity of the document logic and presentation may be tricky in a multi-file and multi-author document preparation scenario. To help, we have laid out the following simple process.

What is a template customization ?

A customization is anything which changes the definition of any of the pre-defined formats, or the addition of new ones. This guide itself suggests a few “standard” customizations which you may choose to adopt, such as:

- variable page heights, Section 1.4, "Page size options";
- iconed paragraphs for impact, Section 2.8, "The zany paragraph formats";
- the cover page, Section 2.9, "The front matter";
- setting the user variable for the running footer to your document title, Section 3.2, "The running footer and background text".

Possibilities of other useful customizations could be: changing few key formats to obtain a distinguishable look and feel for specific document; addition of logical character tags as appropriate for your project, e.g. you may add highlights to denote *Input* or *Output*; addition of logical paragraph tags for formatting special sections, such as function calls (see Section 2.7, "Function calls"). Thus, you may add a *FcnCall* tag, similar to *SideHeadSection*, or *FcnSynopsis*, *FcnParameters*, *FcnInput* and *FcnOutput* tags, all similar to *SideHeadSubSection*.

Managing customizations

All customizations should be centralized into the component file of your reference template (set-up in Section 1.2). This file is then used to update the formats of all similar document files. using *File:Import:Formats...*, as suggested by the illustration. All changes should be logged. The file *chapter.fm* could be used as log file.

Naming convention for new formats

It is a good idea to have a special naming convention for any custom formats that you create, such as postfixing *_xx* to new format names, where *xx* indicates the project being documented. This make locating them later on easier.

Customization checklist

Does this customization:

1. Preserve the visual integrity of the document presentation ?
2. Effect any of the generated lists e.g. the TOC ?
3. Necessitate changes to the configuration for web conversion ? (See Section 3.3)

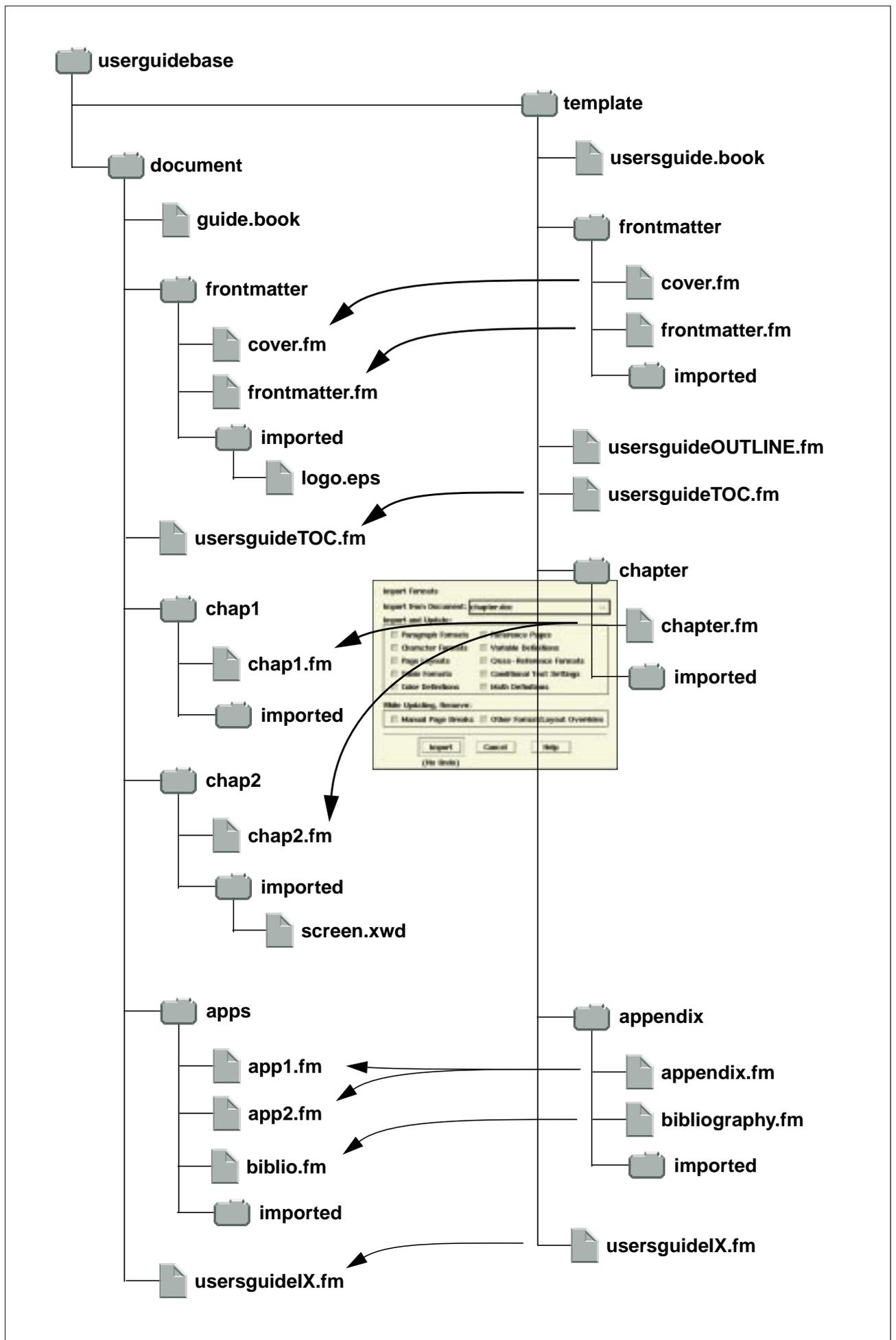


Illustration 1.3 All your customizations should be centralized in your reference copy of the template. The illustration indicates the template master file for each component file in the document. For example, the *Import Formats* dialog is shown to indicate that formats for all chapters are maintained in the chapter template component file.

1.4 Page size options

You may choose to set your document with one of three pre-configured page heights. The paper size however never changes. Thus, any documents produced with different page sizes are still printed on standard paper, be it A4 or US Letter*. Thus, documents with a page height different than that of the paper used will need to be cut after printing or photocopying.

This section describes how you can benefit from these preset options. However, if you desire a page layout not achievable directly with these options, nothing is to stop you from customizing the template page layouts.

Different page sizes to choose from

The default page size is that of the standard paper used, in this case A4 (297mm x 210mm). One may choose to change the height to 254mm, for a less tall document shape, or to 210mm, for a square shaped document. The width does not change in all these cases, i.e. 210mm. Another option is to have the page size as the common denominator between US Letter and A4. The complete list of predefined page sizes is provided in Appendix A.2, "Changing the page layouts for different page sizes".

The different page sizes allow you to produce a more handily-sized document, without investing extra work into the preparation of the layout.

Changing the page size

Page sizes may be changed by importing page layout formats from the format files in the *lib* subdirectory of the distribution. The exact step by step procedure to execute this is provided in Appendix A.2, "Changing the page layouts for different page sizes".

Select the page height as early as possible

Since changing the page height changes also the document styles, the same argument of "Choose the style early" (Section 1.1) applies. Selecting a different page size, or even switching from one to another, is a simple procedure as long as it is done early on. Doing it after most of the document is partitioned for a specific page size, especially in the case of the STOP method, may imply significant amount of manual adjustments and rewriting.

Centralize template customizations

Changing page height is a template customization and thus should be treated as discussed in Section 1.3, "Managing template customizations".

* Default paper size is A4.

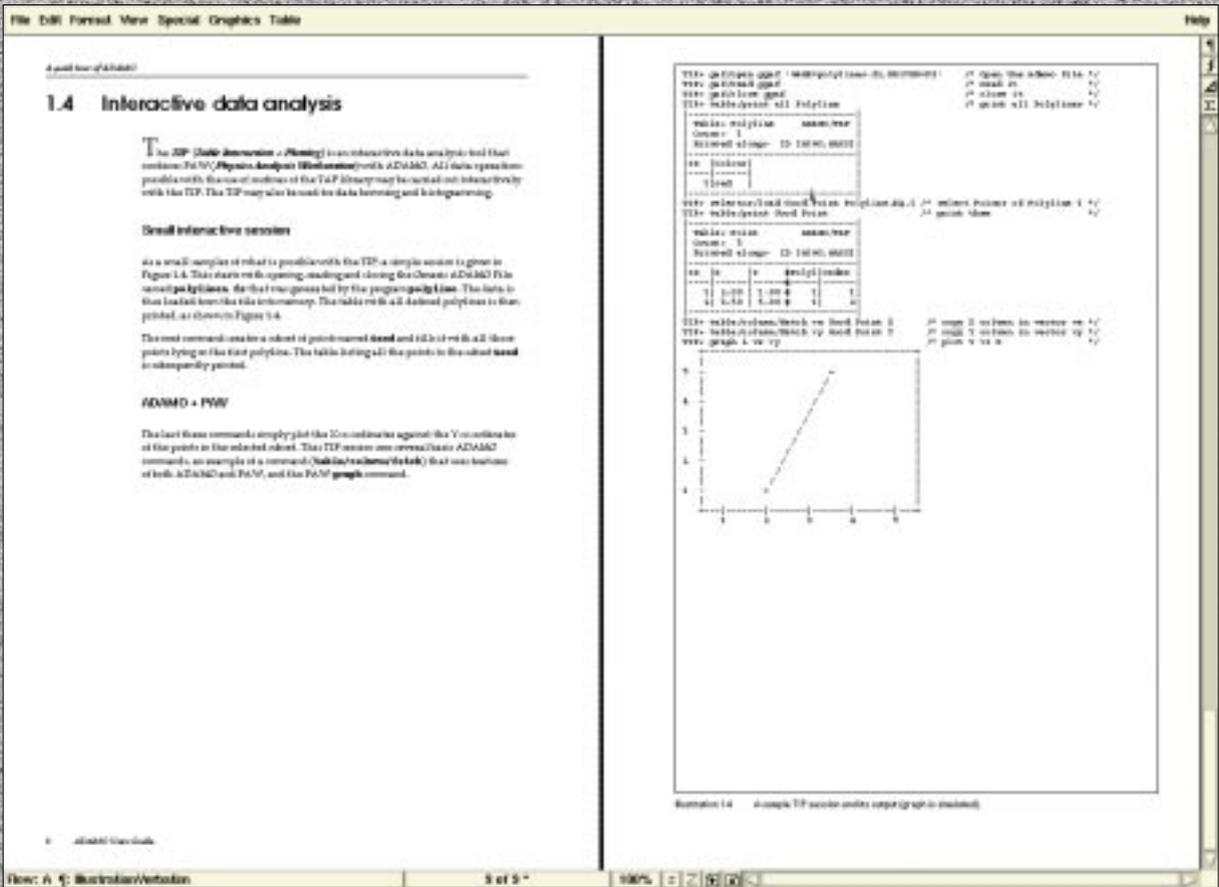


Illustration 1.4 Changing the page height from 297mm (height of A4, upper half of the illustration) to a height of 210mm to give the square shaped pages shown in the lower half of the illustration. An intermediate height of 254mm is also a predefined option.

1.5 Planning your user guide

There are various ways by which you can organise your document, one of which is the task-oriented approach. Other breakdown schemes are possible but it is of fundamental importance that you pick one and sketch out a plan for the book.

The task-oriented approach

The task-oriented approach considers the different tasks that users may want the software to perform, and how they might go about to find out how to do it. This approach helps produce a document that is usable by all types of users, regardless of level of expertise or background. Organizing documents by other criteria, for example by the type of user, may require duplication of a considerable amount of information, such as how to start the system.

Each identified task should be documented in a section that makes complete sense when read separately from the surrounding material. In this way the user does not have to flip pages to complete a basic procedure. Furthermore, basic procedures will not need to be repeated but may be referred to as stand-alone units.

Organize the tasks into chapters

Once you have identified the list of tasks that you would like to document, you should analyze how to best group these tasks into logical chapters and sections. FrameMaker itself may be used for this.

If you do use FrameMaker, before adding chapter files and inserting chapter titles and section headings, it will be more efficient if you first decide on the set-up of your reference template (Section 1.3). This means you would need to also decide which document style to use (Section 1.1), your file organisation (Section 1.2) and the page heights of your document (Section 1.4). If you further customize your reference template you will need to update the formats of any files already added to your FrameMaker book.

Adding files to your FrameMaker book is explained in Section 1.6, "Adding and deleting chapters".

The table of contents

You can get a global view of the content breakdown of the book by updating the table of contents, as detailed in Section 3.1, "Routine book procedures". If you are not happy with the hierarchy or the order of the contents, you can iterate on this procedure using the convenient hypertext feature of the table of contents. To go to a section just click on the corresponding heading entry in the table of contents. Then add, delete, cut, paste, etc. the section headings in the document files before regenerating the table of contents.

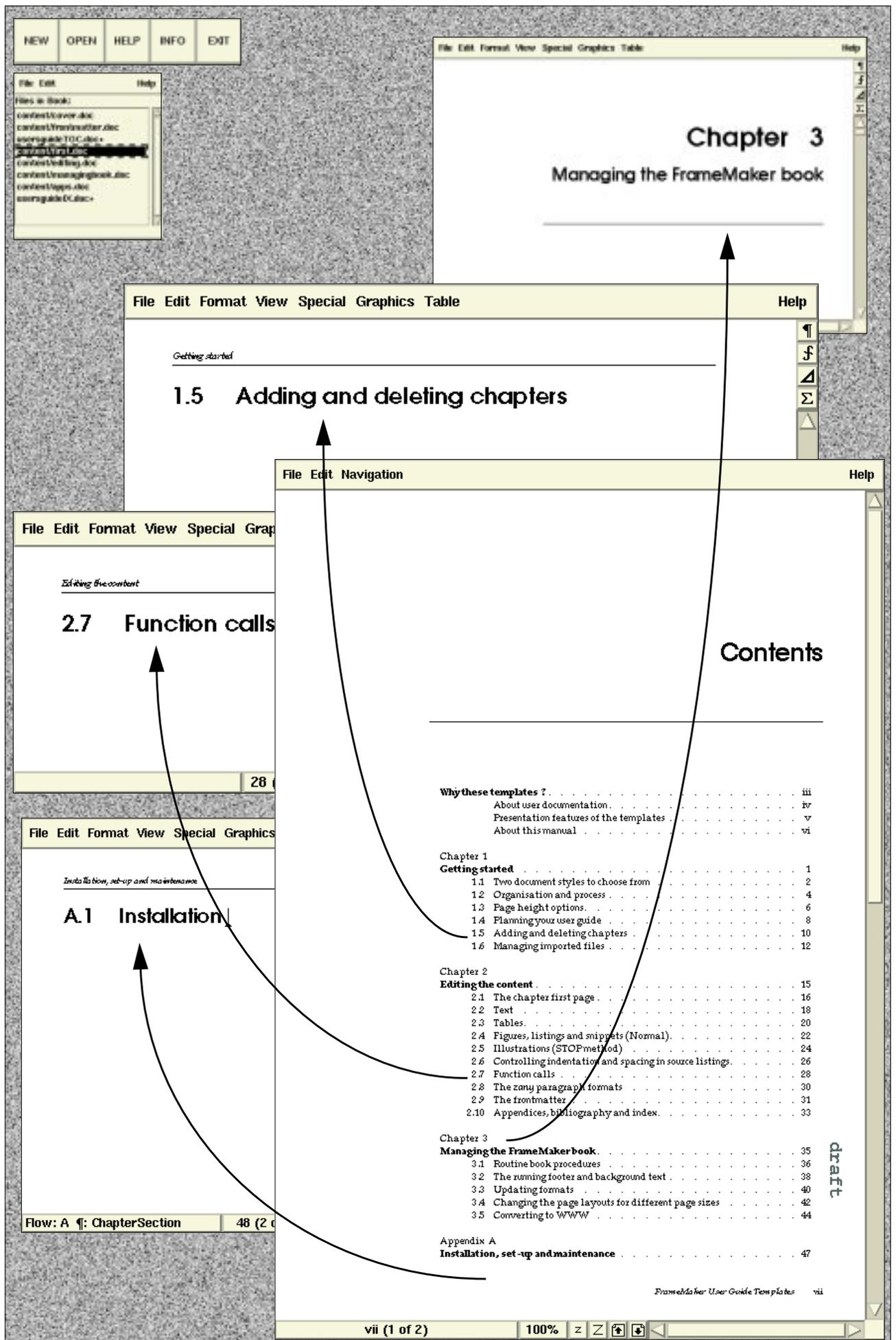


Illustration 1.5 Planning the skeleton for the document using the automatically generated table of contents as a convenient tool to navigate around your document.

1.6 Adding and deleting chapters

Each chapter should be in a separate file, and maybe also in a separate subdirectory. Thus to add a chapter you will need to make a copy of the chapter file template in your document directory structure.

Use your customized template

Before adding chapter files to your FrameMaker book you should have already set-up a reference template which will centralize all your customizations, if any, as explained in Section 1.3, "Managing template customizations". Use the template components of this reference template when adding component files to your book.

Adding content files

Content files can be of type frontmatter, chapter or appendix. In all three cases the way to add files is the same. The procedure for adding is given in the illustration.

You may need to add content files while sketching the table of contents for your document, as suggested in Section 1.5, "Planning your user guide".

Re-adding automatically updated files

Deleting and re-adding the automatically updated files in the book is done in a different way than content files. This is because each of these files has a very specific set-up which needs to be specified at the time when they are added.

For detailed procedure how to re-add each of the automatically updated files see Appendix , "Close all files in the book (Esc fC) by holding down Shift key + clicking on the book File menu to select Close All Files in Book."

Adding new content files to the book

Frontmatter and appendix files are added in the same way.

1. If the chapter is in its own subdirectory, then create one at the same location as the FrameMaker book file for your user guide, using an appropriate name for the chapter to be added.
2. Copy the template chapter file from your customized reference template to where you want the new chapter file to be, assigning it a name of your choice.
3. If the chapter is in its own subdirectory, create an *imported* sub-subdirectory to hold any imported files for the chapter.
4. With FrameMaker open the book file for your user guide.
5. Select *File:Add File...* and browse to the newly copied chapter file.
6. Select the file and set the position in the book where it is to be added.
7. Click on *Add*.
8. Repeat 1. through 7. as many times as necessary to add all the frontmatter, chapter and appendix files that you need.
9. Update the running footer, and any other variables, for the added files as detailed in Section 3.2, "The running footer and background text".
10. Update the book as explained in Section 3.1, "Routine book procedures".

Deleting files from the book

1. From the book file for your user guide select *File:Rearrange File...*
2. In the *Rearrange Files* dialog, click on the file name you want to delete to select it.
3. Click on *Delete*.

Note that this only removes the book reference to this file. If you want to remove it from disk you will need to do so separately.

1.7 Managing imported files

FrameMaker allows you to import both graphic and textual files by copy or by reference. Importing by copy stores a description of the graphic or text inside the FrameMaker file, while importing by reference stores only the path information for the graphic file on the disk.

Advantages of importing by reference

- The FrameMaker document is automatically updated when the graphic or text file changes.
- Less duplication, e.g. if the same image is imported more than once in the same chapter it is not duplicated.
- Keeps the size of the FrameMaker file itself small.
- Large graphics slow down FrameMaker. A file containing many graphics can become very tedious to edit.

When imported by reference, high quality graphics, i.e. with large file sizes, may be temporarily replaced with small lower quality ones. This makes it more comfortable to edit the document while still being able to see and manipulate the graphics.

Advantages of importing by copy

- Guarantees that the FrameMaker file and the graphic stay together, thus the file is easier to move around than a directory structure with relative pathnames.
- Some graphic file formats are not compatible across platforms. When a graphic is imported by copy it is stored internally as a `FrameImage` or `FrameVector` formats which can be displayed on all platforms.



Useful Tips

- If your document becomes slow to edit because of the graphics, you can turn viewing them off completely by selecting *View:Options...*
- To generate a list of all imported files, select *Add File...* from the book *File* menu generate a *List of References* using *Imported Graphics* as the source reference type.
- It is best to keep the number of different formats that you use for imported graphics to a minimum, ideally one for bitmap and one for vector. Many different formats may be inconvenient when you need to process the images or move the document across platforms.

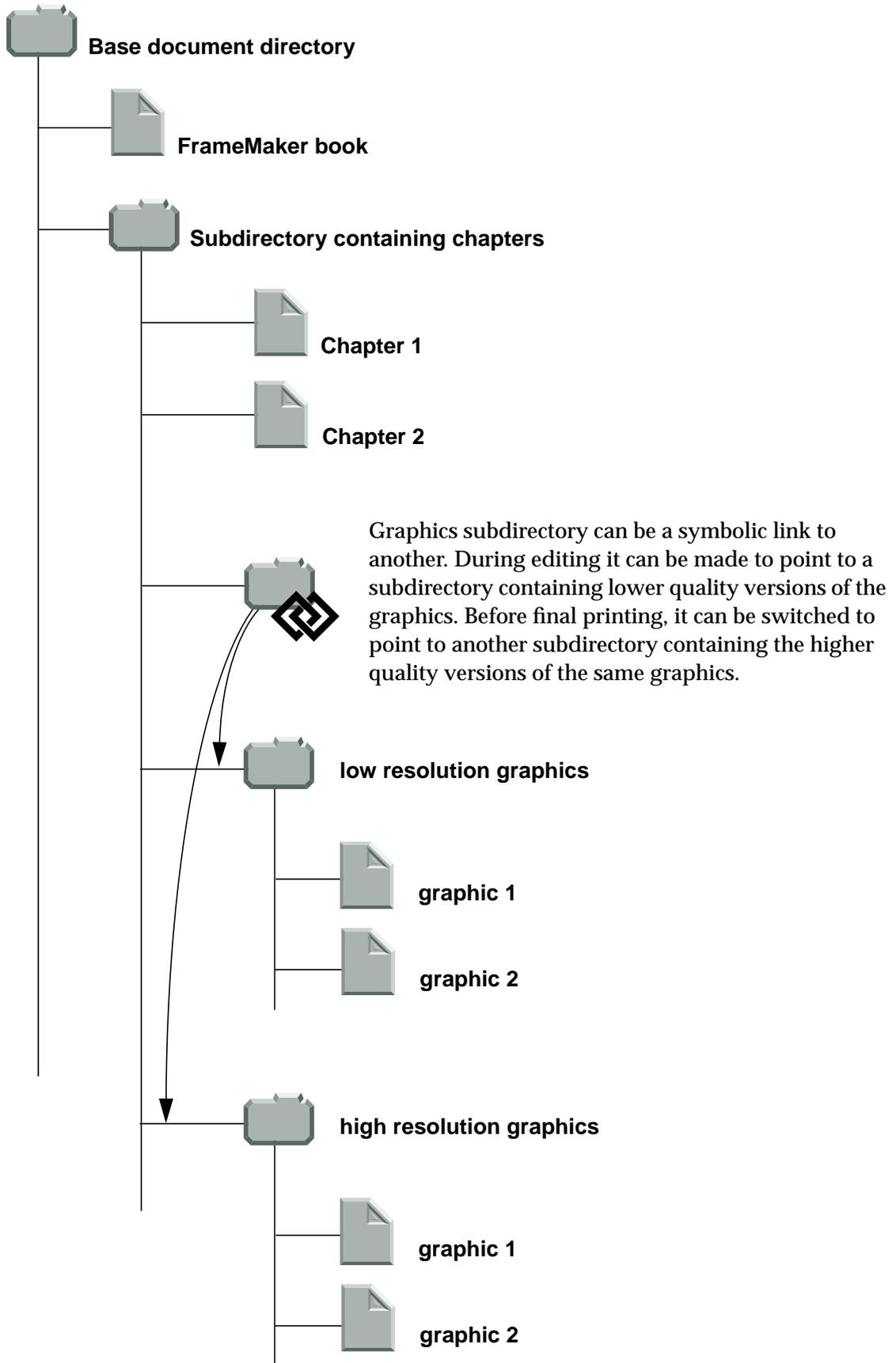


Illustration 1.7 One advantage of importing graphics by reference is that the document can be switched to use either low quality versions of the same graphic files, for quicker editing, or higher quality versions of the same graphics for quality printing.

Chapter 2

Editing the content

This chapter shows you how to format text and images that you either create in FrameMaker itself or you import from other sources. There is a comprehensive set of pre-defined formats to handle chapter and section headings, running text, lists, tables, figures and listings.

2.1	The chapter title page and section headings	18
2.2	Running text	20
2.3	Tables	22
2.4	Figures, listings and snippets (Normal template)	24
2.5	Illustrations (STOP template)	26
2.6	Controlling indentation and spacing in listings	28
2.7	Function calls	30
2.8	The zany paragraph formats.	32
2.9	The front matter	34
2.10	The appendix and the bibliography.	36
2.11	The index.	38

2.1 The chapter title page and section headings

The first page of each chapter is reserved for the chapter title page. It contains the chapter number and title, followed by an introductory paragraph and a local table of contents all top-level sections contained in the chapter. Chapters may have up to three levels of subsections.

The chapter title page

The chapter title page contains a very specific sequence of paragraphs in the following order:

1. 1 *ChapterNumber* paragraph containing the chapter number or, for frontmatter chapters, just blank space;
2. 1 *ChapterTitle* paragraph for the title;
3. 1 or more *Body* paragraphs for the chapter introductory paragraph;
4. 1 blank *Body* paragraph, for spacing;
5. 1 *ChapterSectionList* paragraph for each *ChapterSection* paragraph in the chapter, to make up the table of contents for the chapter. Each will contain a cross-reference to a *ChapterSection* paragraph, using the *ChapterSectionList* cross-reference format.

Sections and subsections

There are three possible sub-levels. The higher level one is *ChapterSection* while the lower ones are *ChapterSubSection* and *ChapterSubSubSection*. The sub-section formats are numbered in the normal report style and not numbered in documents prepared with the STOP method.

The *ChapterSection* tag is set to start on a new page while the sub-section tags start anywhere.

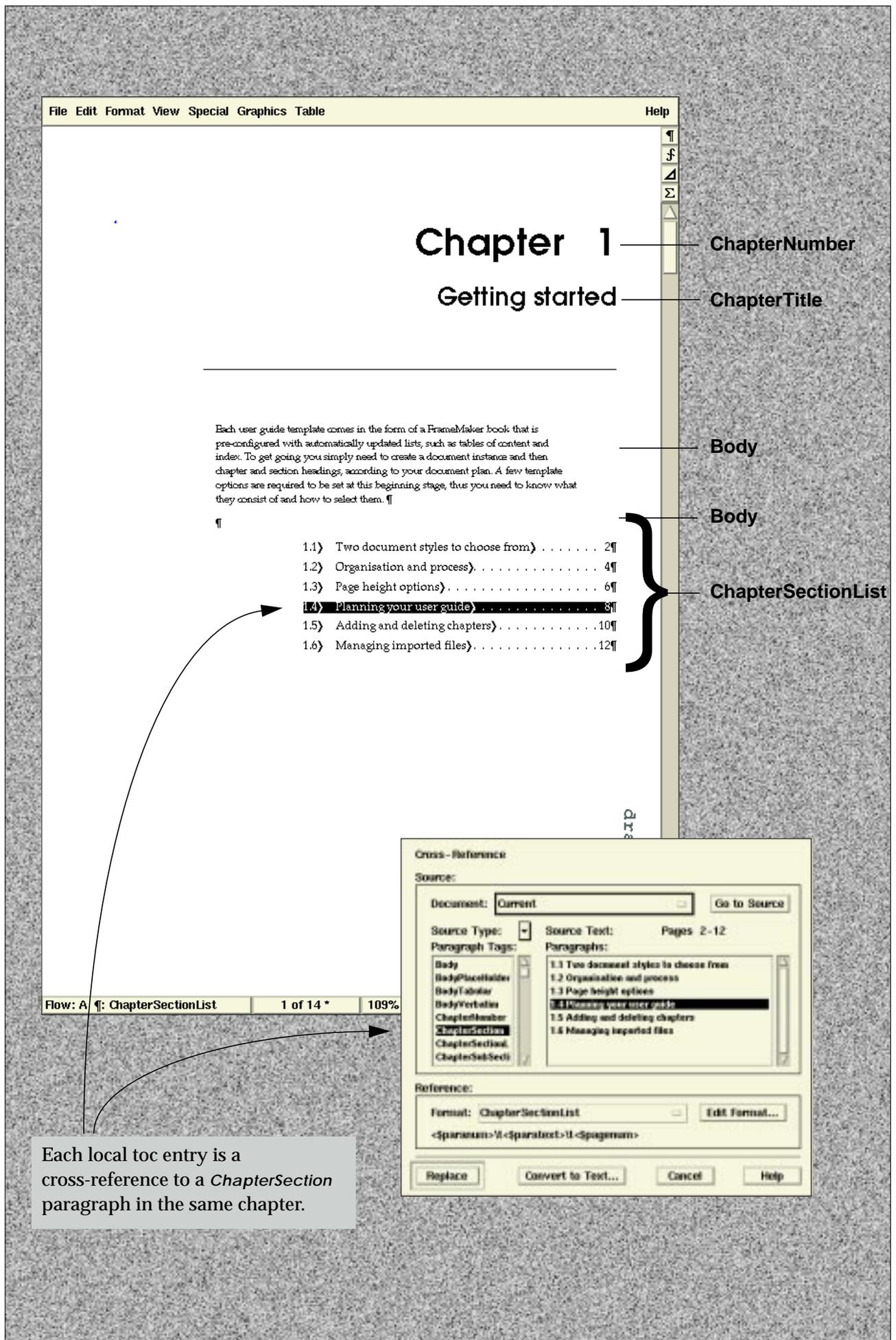


Illustration 2.1 The different paragraph tags that make up the chapter title page.

2.2 Running text

Running text consists of simple paragraphs, bulleted and numbered lists, and hanging indents or definition lists. Running text is formatted using the pre-defined paragraph, character and cross-reference formats. These formats are named such that their use is as self-explanatory as possible.

Paragraph formats

Body The basic paragraph.

BodyPlaceHolder A container paragraph for tables and anchored frames.

BodyTabular Like *Body* but also includes tab stops every 2cm. Can be used to format simple tabular data without using tables.

BodyVerbatim A fixed width style for inserting bits and pieces of code or commands in the running text.

ConceptTerm, ConceptDescription A pair of paragraphs for formatting single concepts that need a title.

DL1Term, DL1Description Hanging indents; a pair of paragraphs for formatting definition lists. Each of these has one or more variations: *DL1TermRIH* is a run-in head, thus the following description paragraph starts on the same line; while *DL1Description<-* is for descriptions that need no indentation and *DL1Description->* is for descriptions that need a lot of indentation.

KeepRestOfPageClear A special paragraph to force empty space at bottom of page.

L1Bullet, L1Hyphen, L1Alpha, L1Number List paragraphs of level 1. For numbered and alphabetic items, there are also special formats to reset numbering. There is a corresponding set for level 2.

L1Continuation, L1ContinuationVerbatim Continuation paragraphs for list items of level 1. There is a corresponding set for level 2.

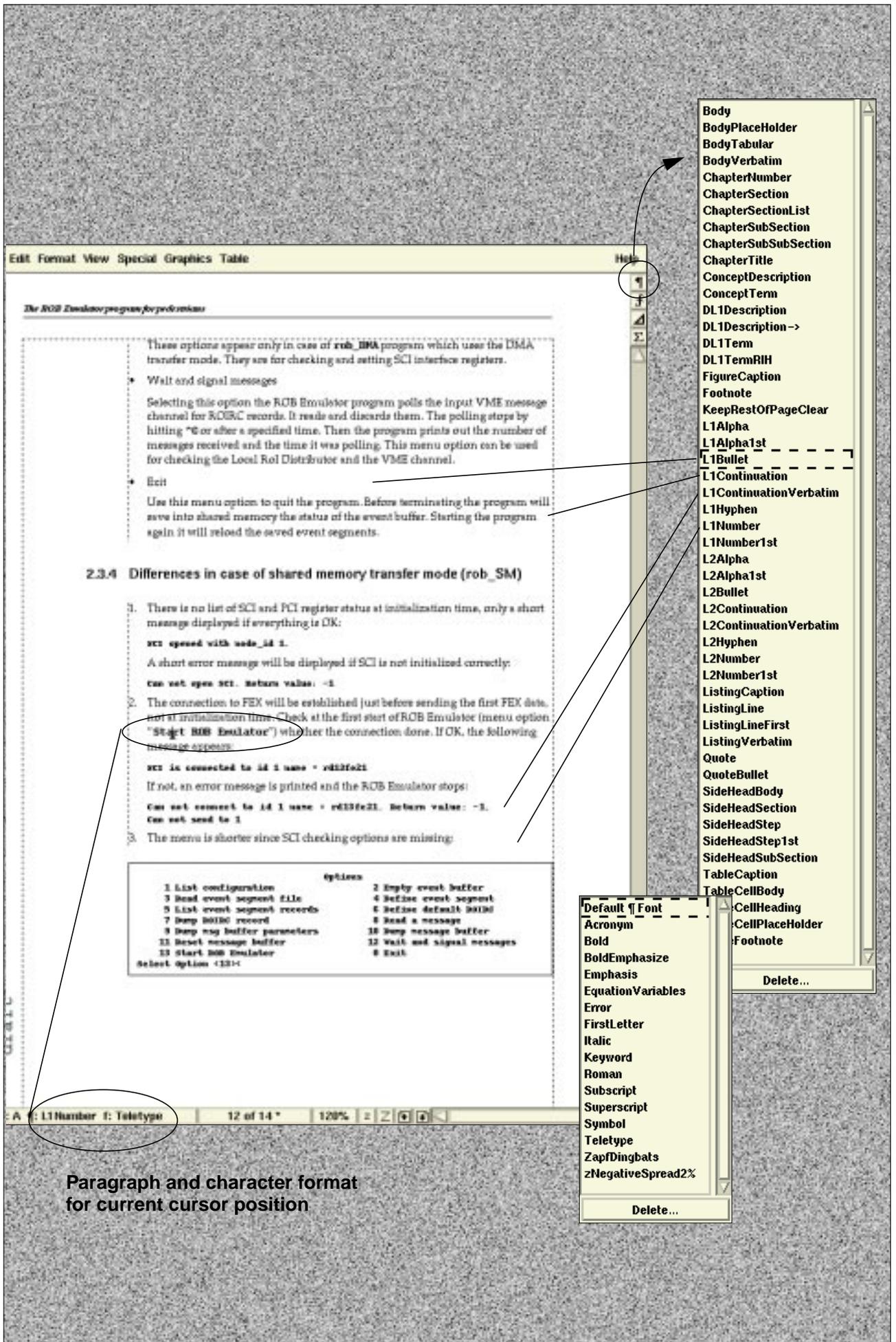
Quote, QuoteBullet For quoted excerpts.

SideHeadStep A paragraph for numbering steps in a procedure. Different from a numbered list as these steps can span different sections. There is another format to reset numbering to 1.

Character formats

General character formats are pre-defined in the template. However it is likely that you will need logical character tags that are more appropriate for your documentation. As an example you may add highlights to denote *Input* or *Output*; Feel free to add useful character tags as necessary.

Adding character is a customization and thus the procedure indicated in Section 1.3, "Managing template customizations" should be followed.



Paragraph and character format for current cursor position

Illustration 2.2 Predefined paragraph and character tags for formatting running text.

2.3 Tables

A few simple table formats are provided as convenient starting points. Properties like straddling and irregular ruling may not be stored as part of a FrameMaker table format in the catalog. There may be a certain amount of table customization necessary.

Inserting a table

1. Create a blank *BodyPlaceholder* paragraph.
2. Select *Table:Insert Table...* and choose one of the pre-defined formats or select one of your own definition.

All tables have an automatically numbered title, *TableCaption*.

Basic pre-defined table formats: *2Col*, *3Col*, *4Col*, *5Col*

Four simple formats for 2-column to 5-column tables, respectively. You will certainly need to customize tables to achieve straddling and irregular ruling. It is advisable that, instead of customizing extensively one of the pre-defined formats, you actually create your own new table formats. If you have many differently customized tables but with the same tag name, you run the risk of losing some of the customizations if you were to update table formats.

If you customize tables try to keep their general look and feel consistent throughout the document. See also Section 1.3, "Managing template customizations".



Useful Tips

- To select cells and columns hold down CONTROL and:
 - single-click to select cell;
 - double-click to select column;
 - triple-click to select entire table.
 - To resize columns select a cell and drag the resize handle.
-



Note

The width of tables should ideally always be the same as that of the main text column text column, i.e. 13.6 cm. You may make it wider if you need to but you should never make it narrower.

2.4 Figures, listings and snippets (Normal template)

Documents prepared using the normal report style may insert figures, listings and snippets wherever necessary in the text. These are all inserted in the same way as tables, as there are pre-defined table formats for each of them.

Inserting a figure

1. Create a blank *BodyPlaceHolder* paragraph.
2. Select *Table:Insert Table...* and choose the *Figure* pre-defined format.

This creates a 1-celled boxed table with an autonumbered *FigureCaption* below it. You may now insert an anchored frame in the single cell to contain your figure.

Listings and snippets

Snippets differ from listings in that they have no caption and, by default, source lines are not numbered. Thus, you use snippets for very short listings of code, commands, source files, etc., that do not need to be captioned or line-numbered.

Inserting

1. Create a blank *BodyPlaceHolder* paragraph
2. Select *Table:Insert Table...* and choose either the *Listing* or the *Snippet* pre-defined format.

In either case this creates a 1-celled boxed table. The default paragraph format in the cell, *ListingLine*, is autonumbered and one per source line should be used, thus producing automatic line numbering. The *ListingLineFirst* paragraph format resets the line count to 1. If you'd rather not have automatic line numbering, you can just use the *ListingVerbatim* paragraph format instead.

Listing tables are automatically created with an autonumbered *ListingCaption*. Snippets have no caption.

Listings longer than 1 page

Do not use the *Listing* table format, as explained above. Instead, just use a *ListingCaption* paragraph followed by sequences of *ListingLine* paragraphs.

The *Listing* table format is there to enclose the listing in a box, improving readability, and to control spacing of the entire unit. However, single table cells cannot be larger than one FrameMaker page, and a box spanning multiple pages cannot be really called a box anymore, can it ?

General description

1.1 Working principles of the two programs

In Demonstrator A and B the ROB's for a subdetector are clustered into *Readout crates* containing several ROB's. Each ROB is connected to the Level 2 system. The role of the Local RoI Distributor is to receive RoI requests and decisions from the Supervisor and transfer decisions to all ROB's and requests to only those ROB's which are involved in the current RoI. Figure 1.1 shows the schematic view of a ROB emulator crate for the "push style" Demonstrator.

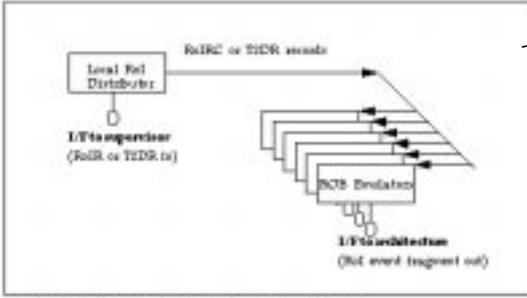


Figure 1.1 Schematic view of the readout buffer crate in demonstrator A and B

The RoI Emulators and

- ROB Emulator
 - Downloading and u
 - Receive RoI Request
 - Maintains buffer size
 - Monitoring
- Local RoI Distributor
 - Downloading and u
 - Receive RoI inform
 - Monitoring

To fulfill the above tasks

1. The ROB Emulators be used to receive RoI of that channel in the Distributor.
2. The Local RoI Distributor itself to send RoI's

RoI Distributor and ROB Emulator for LHC2 B

File Edit Format View Special Graphics Table

4.2 Local RoI Distributor's configuration file

The Local RoI Distributor's configuration file is specified at the program call, e.g.:

```
lrd_exe -s -c lrd.cfg_vieid
```

If only the `-m` option is given then the program tries to find the file named `lrd.cfg` (or `lrd.cfg_000`).

Listing 4.2: The format of the file.

```

1: # configuration file for ROB emulator program
2: # Lines starting with # are comments
3: # Empty lines are ignored
4:
5: NONE      0      # Line not used
6:
7: NONE      lrd_exe  # Line not used, it is a possibility
8: # for defining drive file's name for
9: # the further version of the program.
10:
11: LOOKUP   lrd_barrel_c.Dip  lrd_endcap_c.Dip
12:          lrd_barrel_jet.Dip  lrd_endcap_jet.Dip
13: # Files to define Eta-Phi mapping of
14: # NONE and NONEs. Can specify none
15: # then use, and go to additional line.
16: # It must be a lookup file for every
17: # subdetector and trigger types which
18: # will be used by the demonstrator.
19:
20: SUBDETECTOR THE_BARREL # Which subdetector must match NONE.
21: # Possible subdetectors are:
22: # "NONE", "TRK", "TRK00cap",
23: # "NONE".

```

the NONEs in the crate.

```
conf
T2 setup configuration
on the program find
address and crate_id
address of the NONE.
```

we crate_id and the VME
in file specified by the NET
trator B does not use the T2 setup
he above parameters directly in
he following way:

```
ROB_SPACE
```

he ROB_SPACE command. Several
ME addresses must be given in
with id = 0 has VME start address

5.1.5 Remote control:

Both the ROB Emulator and the Local RoI Distributor programs are prepared to run under a remote control and monitoring system(s). The built-in Minimal Run Control and Monitoring System needs resource registration. A resource consists of a resource identifying number and routine to be called when the resource is activated by the remote control. Includes file `menu.h` defines a structure named `RESOURCE`. It contains a resource number and a pointer to the function to be called when the resource is activated:

```
typedef struct {
  unsigned char resource;
  ctrl_handler handler_func;
} RESOURCE;
```

An array of `RESOURCES` is defined at the beginning of both programs (files `lrd.c` and `rob.c`). To add a new resource item to the program simply add a new member to this array and write the new routine to be called.

At initialization time a call to function `ctrl_init()` initialize the system and defines the server node from where the remote commands are expected. The call to `register_rsrc()` function (from library `menu.c`) make the registration of all resources defined in the resource array. Then the program has to call the routine

Illustration 2.4

Document following the normal report style may insert figures, listings or snippets anywhere in the text. All such objects are anchored in an empty *BodyPlaceholder* paragraph. Line numbering in listings is obtained by using the *ListingLine* paragraph format.

2.5 Illustrations (STOP template)

The template that supports the STOP method* provides pre-defined formatting for full right-page illustrations.

Right-page illustrations

The STOP method defines sections as 2-page spreads, one for text and one for the illustration. The template defines the right page to be the illustration page of STOP sections†. Illustrations may hold graphical informations such as drawings or screen dumps, or textual material such as source code listings.

Formatting illustrations

1. Create a blank *IllustrationPlaceholder* paragraph.

This paragraph is set to start on a new text column thus should be put at the top of the following (*Right*) page.

2. If the master page used is not *IllustrationRight*, then assign this master page to the page where the *IllustrationPlaceholder* paragraph is.

- Select *Format:Page Layout:Master Page Usage...*
- In *Use Master Page:Custom:* select *IllustrationRight*.
- Click *Apply*.

3. With the insertion point in the *IllustrationPlaceholder* paragraph, hit **RETURN**. This will create an *IllustrationCaption* with the same number as the last defined *ChapterSection*.

4. Place the insertion point again in the *IllustrationPlaceholder* paragraph, and, depending on whether the illustration is textual or graphical:

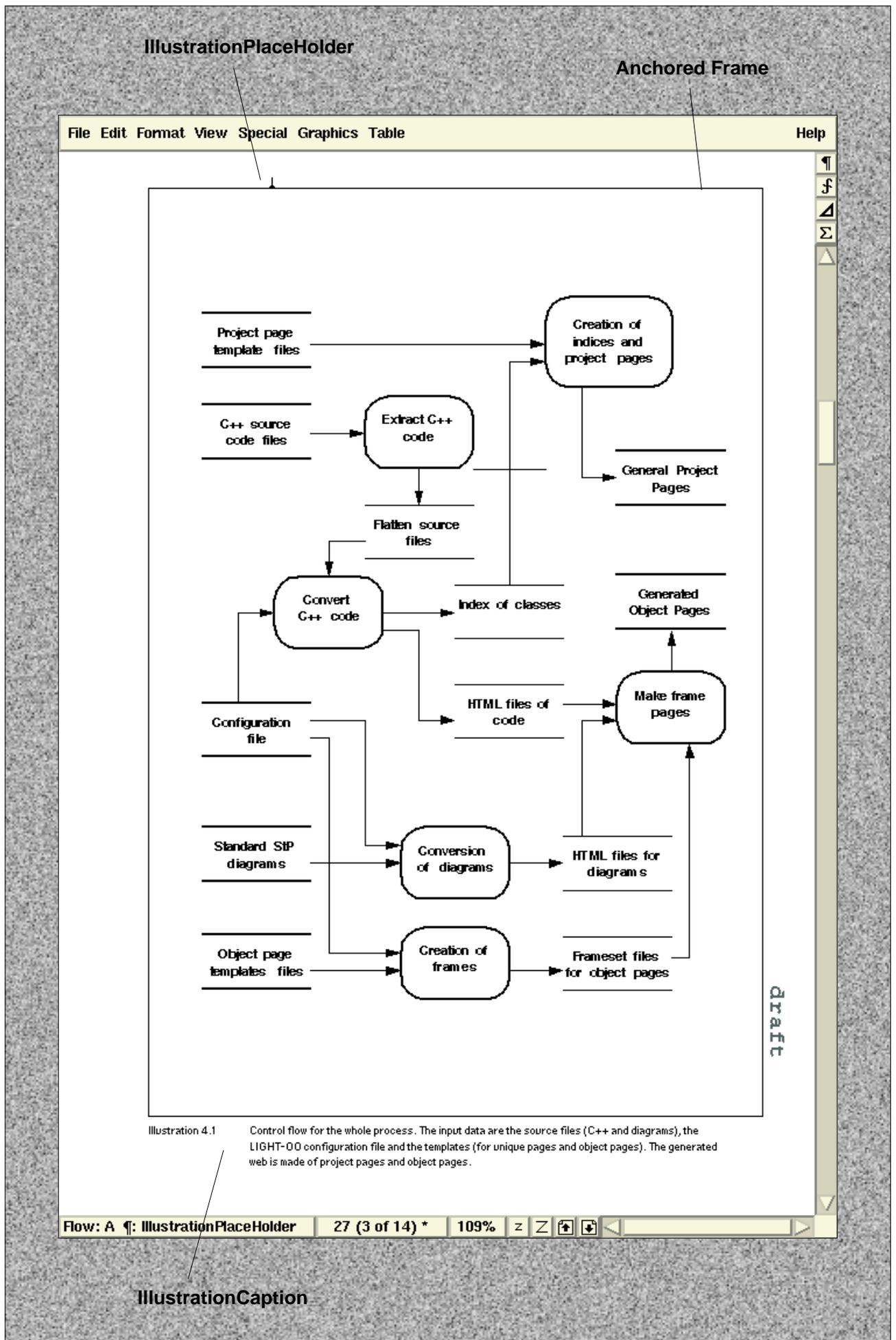
- For textual illustrations insert an *Illustration* table and in it insert the text;
- for graphical illustrations insert an anchored frame with the following properties:

```
Anchoring Position: Below Current Line
Alignment: Center
Width: 16.9 cm
Height: 25.8 cm
```

Into this anchored frame you can now import your graphics or create them with *FrameMaker*.

* See Section 1.1, "Two document styles to choose from".

† If you prefer to have illustrations on the left and text on the right, you will need to modify the autonumbering of the paragraph formats *ChapterSection* and *IllustrationCaption*. Since, in the logical order of the document flow, the first autonumber increment would occur in *IllustrationCaption*, you will just need to remove the "+" from the numbering scheme of *ChapterSection* and add it to the numbering scheme of *IllustrationCaption*.



IllustrationCaption

Illustration 2.5 The various formatting elements of full page illustrations. This example shows a graphical illustration, thus an anchored frame is inserted in an *IllustrationPlaceholder* paragraph. Textual illustration would employ the *Illustration* table format. The size (and border) of the text column for the image, as well as the caption, are specified on the *IllustrationRight* master page.

2.6 Controlling indentation and spacing in listings

By default, the template files have *Smart Spaces*^{*} turned on. This tells FrameMaker to not allow you to insert more than two or more adjacent spaces. In general this is useful but for the special case of code listings it is not, where white space is used to improve clarity. This section describes a few tricks to control your white space in listings.

Inserting multiple spaces

To insert multiple spaces while composing, for example, a code example directly into Frame, use *HardSpaces*, which are inserted with CNTL-Spacebar. You can see the hardspaces on screen by turning on *View:Text Symbols*.

Importing a listing by copying

When you copy a piece of code into Frame, the original white space will normally be retained for you. However, if you then go and edit a line, any sequences of adjacent white spaces will be collapsed to single spaces! To avoid this problem it is recommended that you replace all spaces in imported listings to hardspaces. This is done as follows:

1. Select the entire code listing
2. Pop-up the *Edit:Find/Change...* (or **Esc ef** as keyboard shortcut).
3. Set *Find to Text:* and type a single space in the text field.
4. Set *Change to To Text:* and type a single hardspace in the text field (CNTL-Space).
5. Make sure that *Change All In:* is set to *Selection*.
6. Hit *Change All In:*
7. Accept the *Cannot be undone. OK to continue?* pop-up.

Importing a listing by reference

When importing text files by reference, any white spaces in the source are retained. The only adjustment you are allowed to do is to assign an appropriate paragraph tag to the entire inset, e.g. *ListingLineFirst* or *ListingVerbatim*.

Tabular characters in the source file are problematic as tabs in Frame have a different meaning than tabs in ascii files. It is better to replace any tabs in the source file with the appropriate spaces. If this is not possible, then I suggest you create a special paragraph format to simulate the same pattern of tab stops in the source.

* *Format:Document:Text Options...*

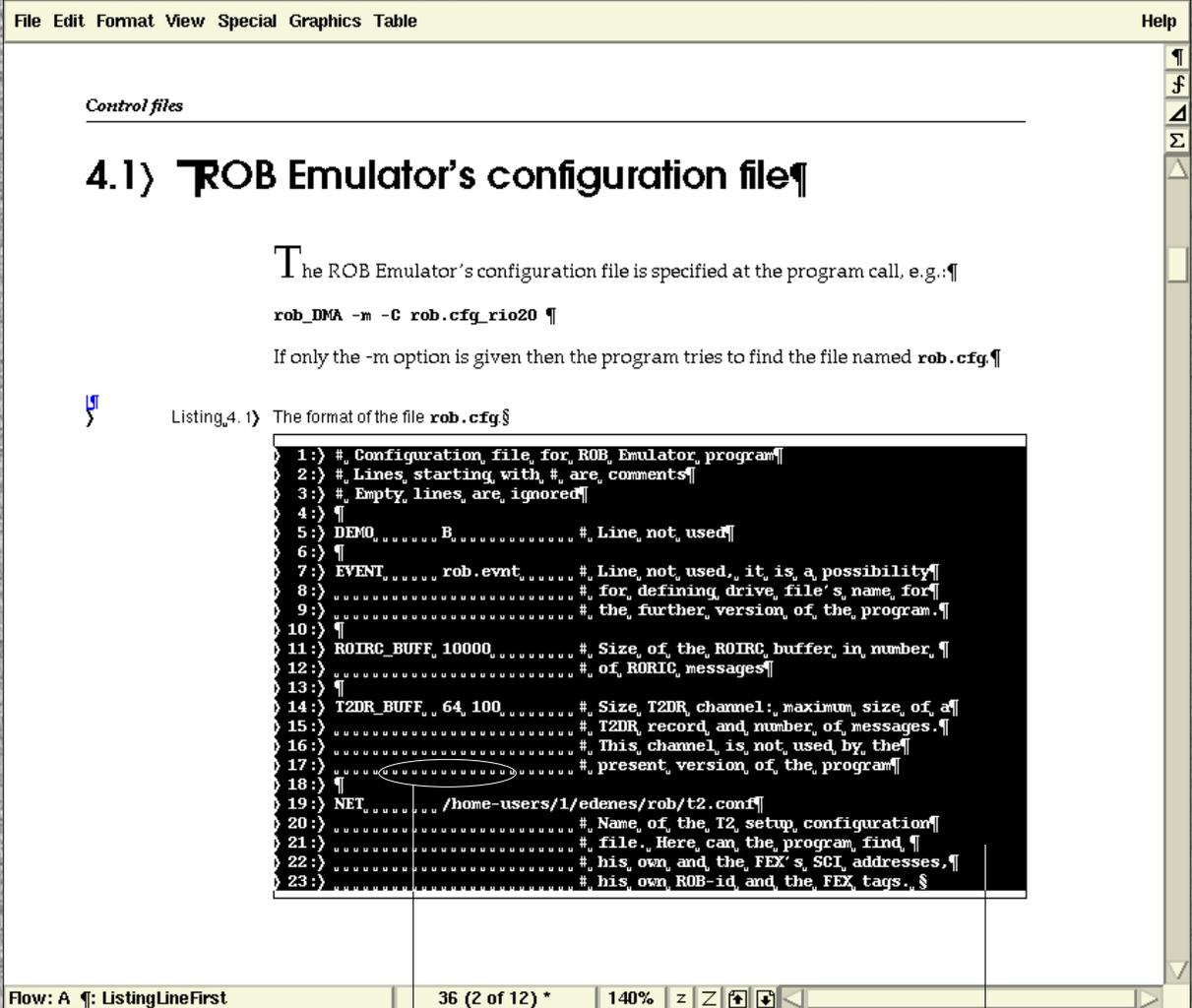


Illustration 2.6 Controlling white spaces in listings by replacing them with hardspaces. In this example text symbols are turned on.

2.7 Function calls

Documentation for such things as function calls is more appropriate in a reference manual. However, pre-defined paragraph formats are available in case you have the need to include such documentation units into your user guide.

Formatting separate units such as functions

Two special paragraph formats have been added to allow the notion of having a separate block of text with its own subsections. These can be used in conjunction with all other running text formats for separate units such as function calls. The two formats are:

SideHeadSection For the title of such sections, such as the actual function call. A line is created above the title to separate from previous content.

SideHeadSubSection For the subsections of these separate units, such as specification for input and output, error messages, etc. The short headings appear right-aligned in the sidehead column space of the document.



Note

The *SideHeadSection* paragraph format does not appear in the automatically updated table of contents.

Customizing for more specific paragraph tags

You may derive new logical paragraph styles to tag special sections such as functions calls. Thus, similar to *SideHeadSection*, you could add a *FcnCall* tag, and similar to *SideHeadSubSection*, you could add *FcnSynopsis*, *FcnParameters*, *FcnInput* and *FcnOutput* tags.

Any customizations should be managed as recommended in Section 1.3, "Managing template customizations".

2.8 The *zany* paragraph formats

Clarity and impact of special pieces of information are sometimes improved if they are associated with an easily identifiable visual image. This information could be things such as notes, warnings, summaries, hints, bugs, etc. These templates provide you the option to import a selection of special *zany** paragraphs into your document, each of which is associated automatically with an icon.

If the predefined set is not enough, an easy formula for adding your own is provided. An extensive list of graphic files for sample icons is also included, to which you can also add your own creations.

Use only few *zany* paragraph formats

You should only use up to around four different types of *zany* paragraphs in a given document—using too many will undermine the effect of clarity and directness that you want to achieve. Also, the size of each FrameMaker file increases by about 1K per *zany* paragraph format. Thus it is recommended that you simply copy the ones that you need rather than import all the predefined formats.

Importing *zany* formats or creating new ones

Adding these formats to your document should be managed as any other template customization, as discussed in Section 1.3. The exact procedures to create and add are described in Section A.5, "Copying and creating Zany paragraphs".

Using *zany* formats

Zany paragraph formats are applied just like any other pre-defined format in the paragraph catalog. The icon, the paragraph title and the lines above and below are part of the format and defined as FrameMaker *Reference Page Frames*.

Zany sections

In the less common case of emphasizing a sequence of several paragraphs, a different technique is used. First insert a table of the pre-defined *ZanyTextSection* table format, just as you would insert any other table (see Section 2.3). Then:

1. Re-tag the single *Zany* paragraph in the left cell to the desired one.
2. Type in the heading and text for the section can into the right cell.
3. Using Table:Custom Ruling and Shading assign a Very Thin line to Top and Bottom of right cell.

* The word *zany* derives from the Italian word *zanni* for a traditional masked clown. As an adjective, it now means something fantastically or absurdly ludicrous. Also, since it starts with the letter Z, FrameMaker lists all *Zany* formats at the bottom of the paragraph catalog.

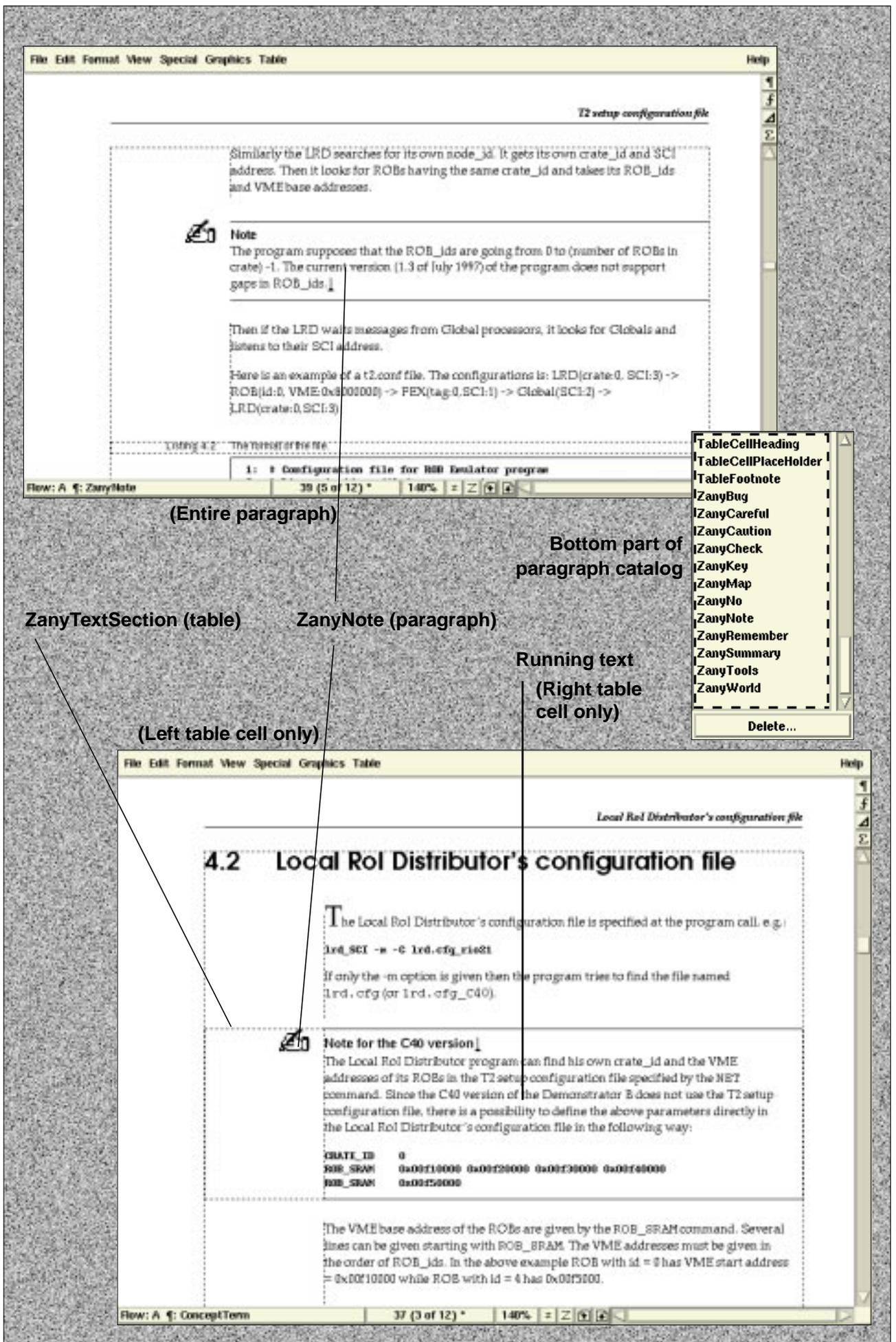


Illustration 2.8 Example of single zany paragraph and of a multiple-paragraph zany section, encapsulated in a *ZanyTextSection* table, as indicated by the visible borders. Shown also is the bottom part of the paragraph catalog showing a selection of predefined zany paragraph formats.

2.9 The front matter

The front matter consists of the cover page as well as the introductory material, such as the preface, prior to the first chapter. This also includes any automatically updated files, such as the table of contents, but these are never manually edited so are not discussed here.

The cover and inside cover

The cover is not only the first impression of your document that users get., but is also the thing that makes your document be recognizable and stand out from a pile of other documents. Thus, the way the cover looks is quite important.

There are pre-defined formats to make a simple cover page quickly, but you may need to play around and change formats to achieve a look that best fits your document. Probably you will only need to modify paragraph formats and the *Right* master page of the file *cover.fm*. Managing template customizations is discussed in Section 1.3.

The inside cover contains the colophon and a generic trademark disclaimer. You may choose to add other statements, such as a copyright statement, or change what is provided as long as the reference to the template source remains.

Editing the frontmatter file

Editing the frontmatter file is identical to editing chapter files except that there is no paragraph numbering. Thus everything else discussed for chapter files—such as the chapter first page, running text, figures and tables—is applicable also to frontmatter files.

Roman page numbering

Page numbering of the frontmatter files should have roman page numbering while the rest of the document should have arabic numerals. If, for some reason, page numbering in a frontmatter file is not roman, it can be reset by

Format:Document:Numbering...

The numbering should be restarted at arabic 1 with the first chapter.

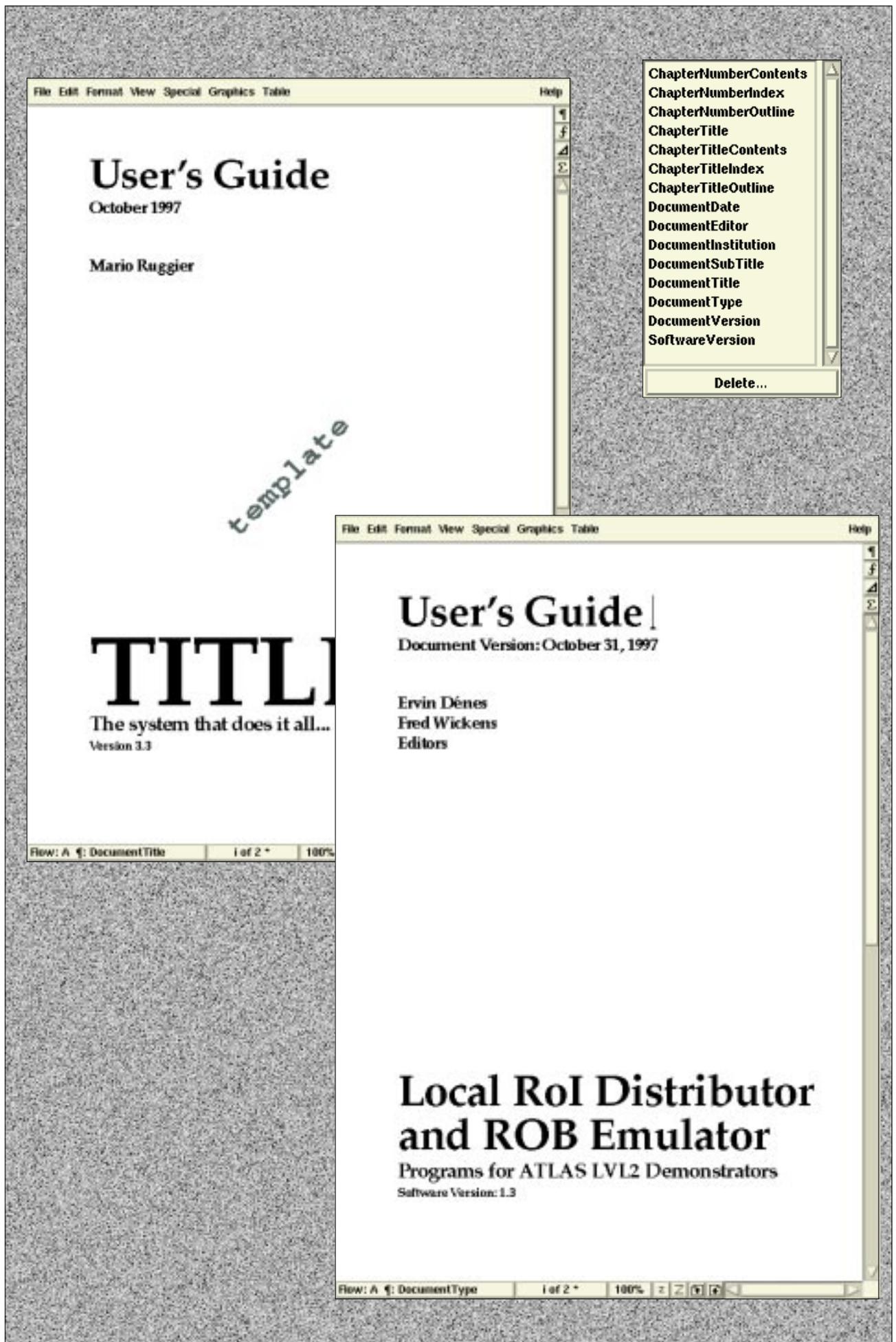


Illustration 2.9 The formats provided by the template for the cover page may need to be modified to achieve the appropriate look for your document, as shown by the difference in the paragraph formats of the default template cover and the sample document cover above. Also shown is the paragraph catalog for the cover file, listing the available styles for the cover page.

2.10 The appendix and the bibliography

The appendices

The appendices are just chapters that use the appendix template file. Everything said previously about editing chapters also applies to appendices. The only difference being that paragraph numbering in the appendices uses an alphabetic scheme.

The bibliography

The bibliography should follow the appendices. Entries are formatted using pairs of paragraph formats and can be grouped into sections:

BiblioItem Contains the document title, identified by a number.

BiblioBody Contains the additional information for the title, such as the authors, publishing house, year, etc.

BiblioSection To group entries into logical blocks (not included in the TOC).

References to bibliography titles

References to publications listed in the bibliography are nothing more than cross-references to *BiblioItem* paragraphs in the bibliography file, using the *Bibliography* cross-reference format.

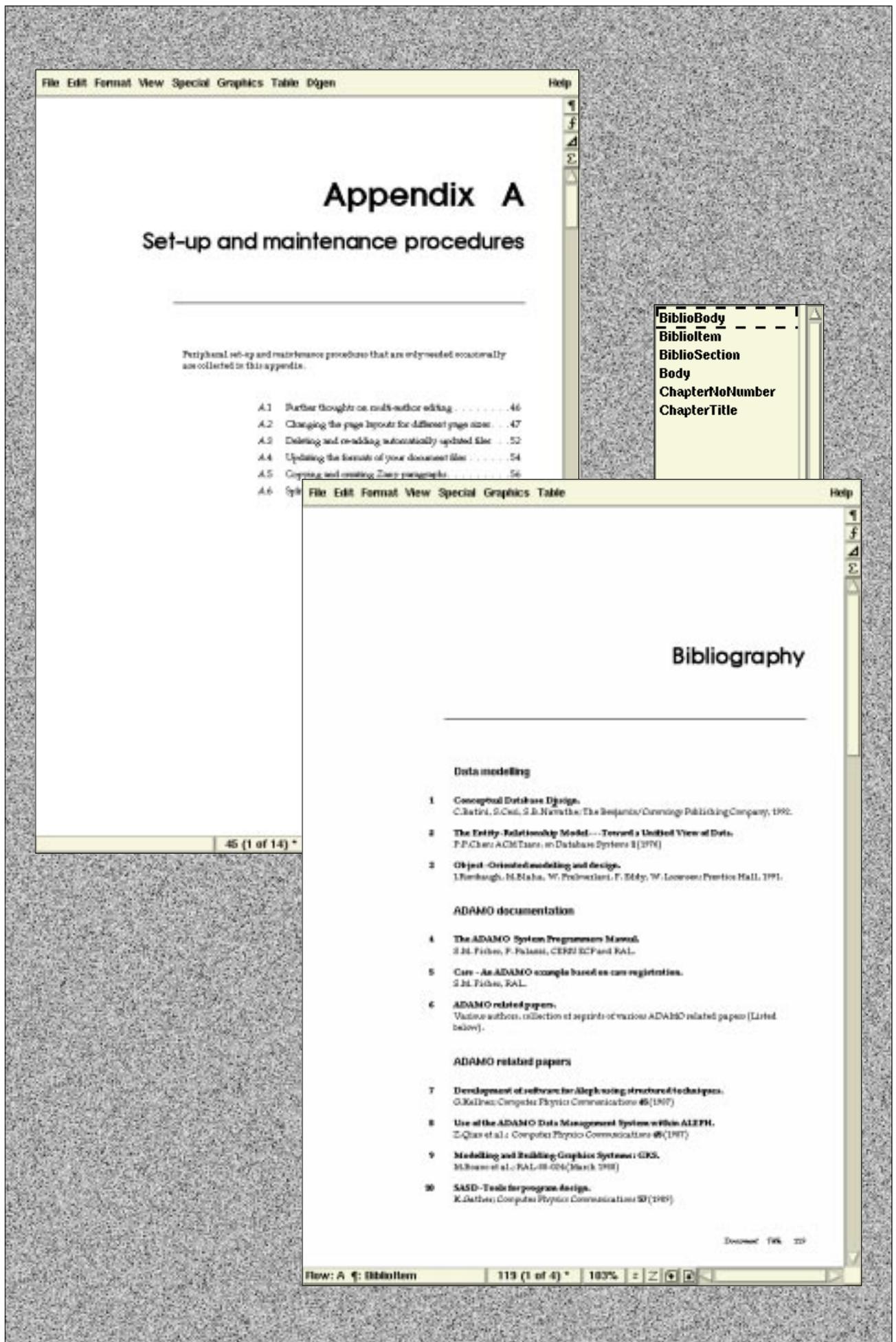


Illustration 2.10 Examples of an appendix first page and a bibliography. Also shown is the minimal paragraph catalog for the bibliography, as opposed to the one shown in Section 2.2, "Running text". Keeping dedicated paragraph formats in the corresponding template files helps keep the each template file type less cluttered and easier to use.

2.11 The index

“A document without an index has been likened to a country without a map, and it is generally accepted that nearly every work of non-fiction is far more useful if provided with such a chart in the form of an index.” [1]

Importance of the index

The index is the user’s most important key for finding information in any document that is very likely to be read in a random access fashion. This category of documents definitely includes user guides and reference manuals.

One of the most common complaints about technical documents is the lack of an index, or a poorly designed index [1]. A good index adds value to any book:

- Users can find specific information efficiently.
- Users decide whether a manual is going to be useful or not depending on what they find, or do not find, in the index.
- Some users will outrightly refuse to use a manual unless it contains an index.

Building the index

The job of the indexer is to make sure that all relevant statements in a manual are recorded in the index in such a way that the user can find them easily. Important issues are conciseness, extent of indexing (as many as 10 per page), appropriate use of logical headings and *see* or *see also* cross-references.

To add index entries to your FrameMaker document you insert index markers in the text throughout the document. The markers are then automatically compiled to build the index file which therefore should never be edited by hand. Index markers have a specific syntax, the basic building blocks of which are shown in Table 2.1.

Table 2.1 Basic building blocks for index entries. For complete information and examples of index entries consult the FrameMaker documentation.

Building block	Effect
: (colon)	Separates levels in an entry
; (semicolon)	Separates entries in a marker
<\$npage>	Suppresses the page number in the entry
<CharacterTag>	Changes character format in the entry. To change back to default you would need insert <Default Para Font>.

Additional tools exist to facilitate the creation and management of index markers in a FrameMaker book. One such tool is IXgen from Frank Stearns Associates [5]. Please ask us about IXgen’s availability at CERN.

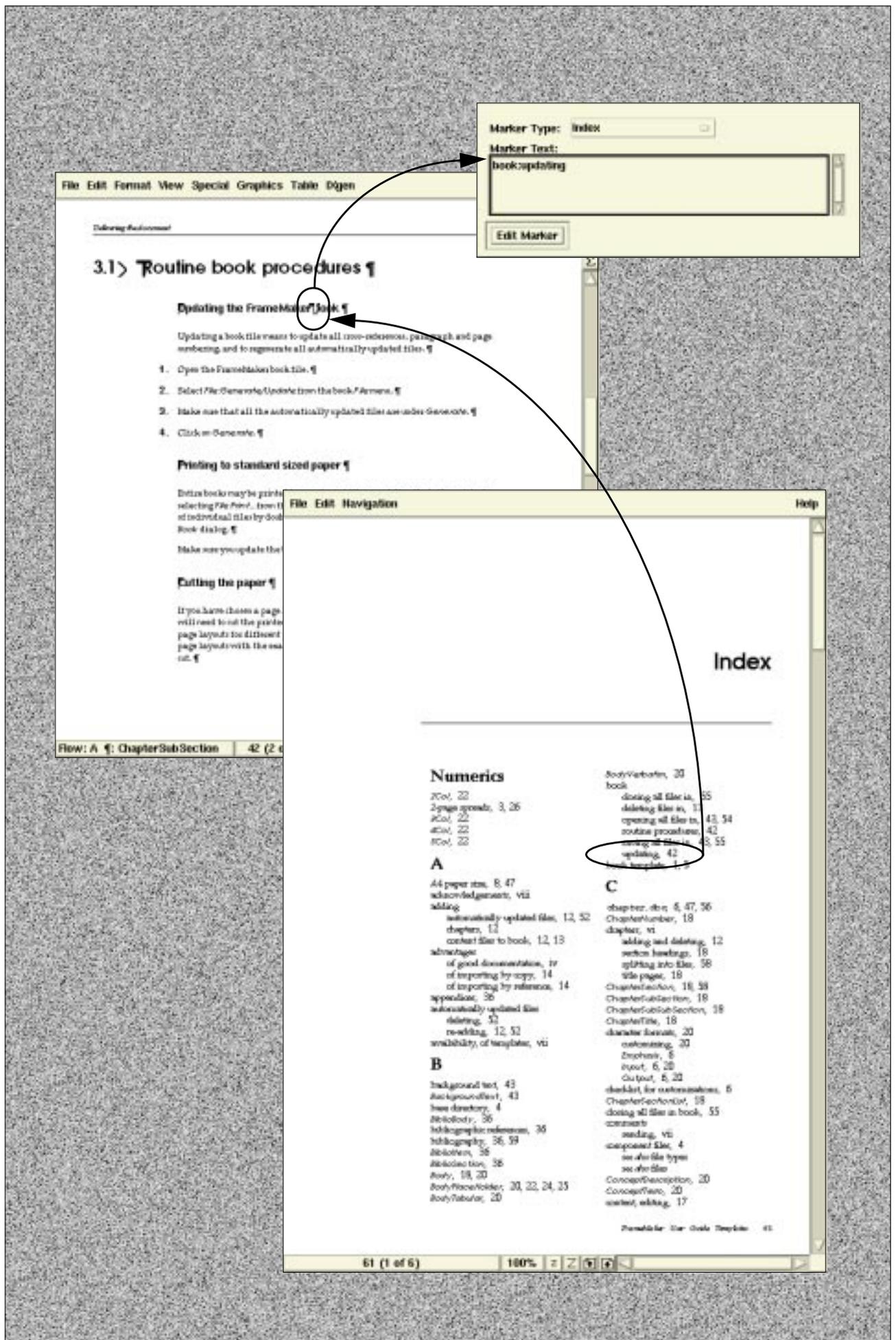


Illustration 2.11 Index entries are created by inserting index markers in the text. When *Text Symbols* are on, index markers appear as a **T**. As the index file is a locked for hypertext (menus are different), it can be used as a document navigation tool in a similar way as the table of contents (see Section 1.5). The marker window shows the text contained in the selected marker.

Chapter 3

Delivering the document

To produce the printed or WWW versions of the document you will need to perform a few extra procedures, such as updating and printing.

3.1	Routine book procedures.	42
3.2	The running footer and background text	43
3.3	Converting to WWW	44

3.1 Routine book procedures

Updating the FrameMaker book

Updating a book file means to update all cross-references, paragraph and page numbering, and to regenerate all automatically updated files.

1. Open the FrameMaker book file.
2. Select *File:Generate/Update* from the book *File* menu.
3. Make sure that all the automatically updated files are under *Generate*.
4. Click on *Generate*.

Printing to standard sized paper

Entire books may be printed to a postscript file or sent directly to the printer by selecting *File:Print...* from the book window. You may selectively suppress printing of individual files by double-clicking on them in the *Print*: column of the *Print Files in Book* dialog.

Make sure you update the book, as explained above, before printing.

Cutting the paper

If you have chosen a page height that is smaller than the standard paper size, you will need to cut the printed or photocopied output. Section A.2, "Changing the page layouts for different page sizes" gives a complete list of the sets of predefined page layouts with the exact measurements, and how the printed output should be cut.

3.2 The running footer and background text

The running footer, normally showing the main title of the document, and the background text on the outer edge of the page, such as “draft”, may be controlled through the two predefined FrameMaker user variables *DocumentTitle* and *BackgroundText*.

These variables can first be reset in a single file of the book and then propagated to all other files of the book.

Resetting the variables in a single file

1. Click into a text column.
2. Select *Special:Variable...*
3. Use the *Edit Definition...* dialog to change the value of the either the *DocumentTitle* or the *BackgroundText* user variables.

Propagating to all other files in the book

1. Make sure that the FrameMaker book file is open.
2. Open all files in the book (**ESC** **FO**) by holding down **SHIFT** key + clicking on the book *File* menu, and selecting *Open All Files in Book*.
3. Select *File:Import:Formats...* from the book *File* menu.
4. Under *Import from Document*: select the file which contains the correct variable settings.
5. Make sure that all formats *except* for *Variable Definitions* are de-selected.
6. Make sure that all files are under *Update*.
7. Click on *Import*.
8. Update the book as explained in Section 3.1, "Routine book procedures".
9. Save all files in the book (**ESC** **FS**) by holding down **SHIFT** key + clicking on the book *File* menu, and selecting *Save All Files in Book*.



Note

In the above procedure, opening all files in the book and updating the book serve as a workaround to a FrameMaker bug with updating locked hypertext files, e.g. the table of contents and index of these templates.

3.3 Converting to WWW

The book template is pre-configured for saving as HTML, using the built-in *Book File:Save As... HTML* module of FrameMaker. This configuration assumes that entire books, as opposed to single individual files, are saved as HTML. However, some document-specific individualisation is required, both before and after saving as HTML. These steps are explained below.

Another possibility as a format appropriate for distribution on WWW is PDF, supported by the new FrameMaker 5.5 *Save as PDF...* feature, explained below.

Before saving as HTML...

Go to the *BookHTML* reference pages (*View:Reference Pages*), locate the *Systems Macros* table, and modify the following information, as necessary:

1. The context links in macros *StartOfDoc*, *StartOfSubDoc* and *StartOfLastSubDoc*. The HTML code, shown below, is inserted at the top of every generated HTML page. You may wish to change it.

```
<small>
<a href="http://www.cern.ch/">CERN</a> ->
<a href="http://www.cern.ch/IPT/">IPT Group</a> ->
<a href="http://www.cern.ch/FrameMaker">FrameMaker at CERN</a>
</small>
```

2. The title text for the document, in macros *StartOfDoc*, *StartOfSubDoc* and *StartOfLastSubDoc*, included in every generated HTML page. The initial value is shown below:

```
<small>PROJECT User's Guide</small>
```

3. The signature of each generated HTML page, in macros *EndOfDoc*, *EndOfSubDoc* and *EndOfLastSubDoc*. In particular you may wish to change the date, the name of the webmaster, and the URL this is linked to, if any. The initial setting is:

```
<address>
2 July 1998 -
<a href="http://consult.cern.ch/xwho/people/XXXXX">WebMaster</a>
</address>
```

After saving as HTML...

After each time you save as HTML, you are required to (in the same directory as the generated HTML files):

1. Over-write the generated cascading style sheet file (to be found in the same directory as all the HTML files) with the one provided with the template, *shared/ug-master.css*. E.g. (UNIX):

```
cp -p <template>/shared/ug-master.css ug.css
```

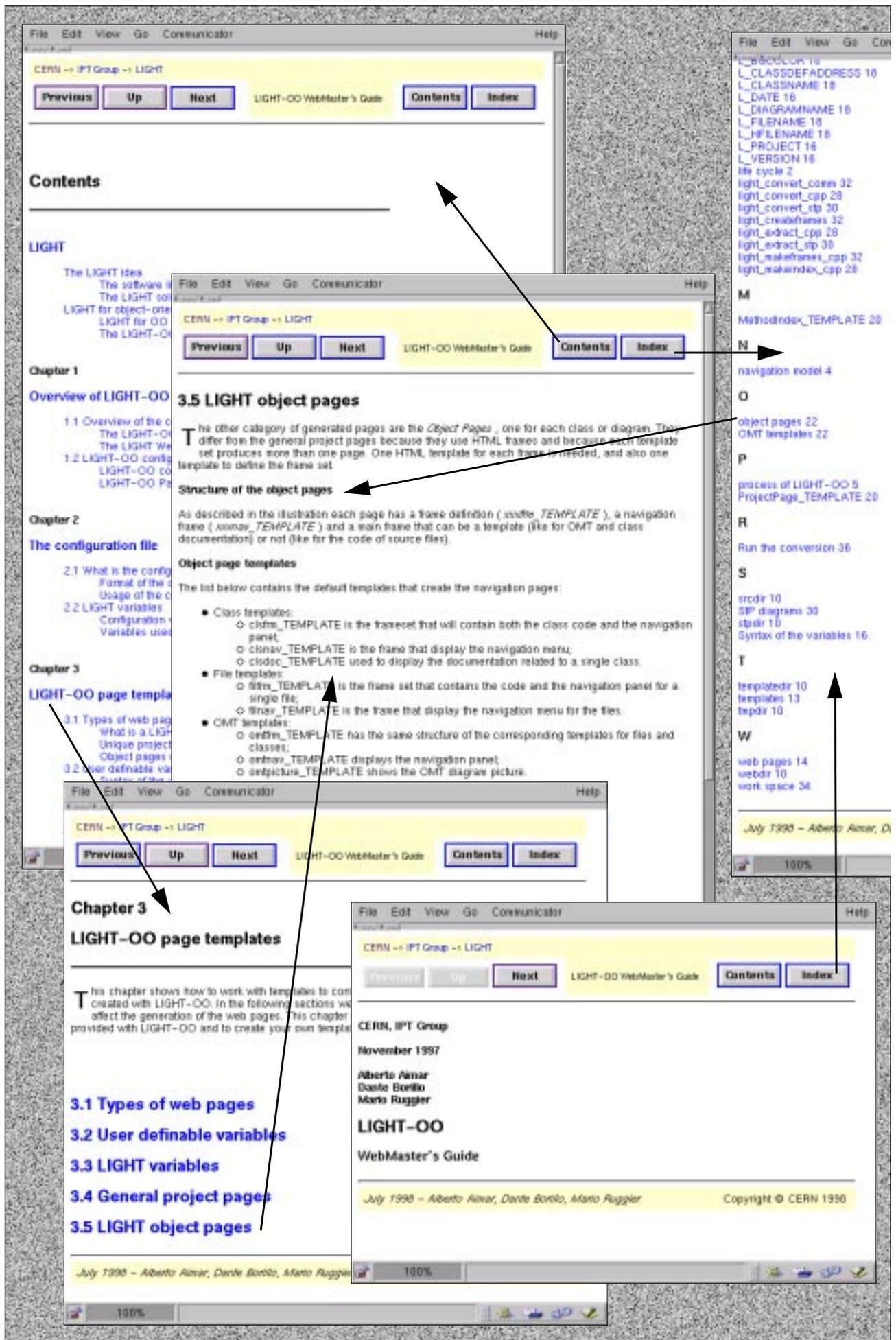


Illustration 3.3 Example of an automatically generated structured web. Navigation aids include next/previous buttons for linear navigation, up/top buttons to quickly go up the document tree, and global and local tables of content, as well as the index, for jumping to specific pages.

2. Create symbolic links *contents.html* and *index.html* that point to the HTML files with the global table of contents and index, respectively. E.g. (UNIX):

```
ln -s ug.4.html contents.html
ln -s ug.1e.html index.html
```

Alternatively, just copy the files:

```
cp -p ug.4.html contents.html
cp -p ug.1e.html index.html
```

where *ug.4.html* and *ug.1e.html* are replaced with the actual names of the generated HTML files for your TOC and index.

3. Create a symbolic link *welcome.html* that points to the first file of the generated web. E.g. (UNIX):

```
ln -s ug.html welcome.html
```

Or, alternatively, just copy the file in the same way as in 2.

Saving as PDF

Saving as PDF is straight forward, except for the setting up of bookmarks. It is recommended that the same hierarchy as the FrameMaker table of contents is used. Thus, the list below *Include Paragraphs*: of the *Acrobat Setup* dialog should contain all heading paragraph tags, using the same hierarchy as in the generated table of contents (see Appendix A.3, "Deleting and re-adding automatically updated files")

To guarantee that hyperlinks within the document work properly in PDF, it is better to produce one PDF file for the entire document, as opposed to single ones for each source file.

Dealing with changes to the template

Section 1.3, "Managing template customizations" discusses a systematic way to manage customizations to the template. For each customization you make to the template you will need to update the *BookHTML* and the *BookHeadings* reference pages accordingly. Documentation is provided by the Adobe FrameMaker User Guide [4].

Appendix A

Set-up and maintenance procedures

Peripheral set-up and maintenance procedures that are only needed occasionally are collected in this appendix.

A.1	Further thoughts on multi-author editing	48
A.2	Changing the page layouts for different page sizes .	49
A.3	Deleting and re-adding automatically updated files	54
A.4	Updating the formats of your document files	56
A.5	Copying and creating Zany paragraphs	58
A.6	Splitting a chapter into several files	60

A.1 Further thoughts on multi-author editing

The main editor should decide how to orchestrate the multi-author writing effort. In addition to file organization and ownership, there are the issues of version control, submitting of material by authors to the editor and the handling of cross-references between files owned by different people.

Version control of the entire document

Any configuration management tool may be used for version control. However, if this implies too much overhead then at least a simple scheme should be put in place to archive a complete copy of each versioned release of the document.

Author submissions of material in progress to editor

The problem with material in progress that is passed between author and editor is to keep clear track of which copy is the current master and who owns it.

The editor should clarify how chapters, i.e. subdirectories of files, are to be submitted for compiling a new version of the book, and for possible format updates. Ideally this is done by a check-in and check-out mechanism, implying that after a chapter has been checked in, the author of that chapter must not continue to work on his copy but should wait for the editor to release the updated version of the chapter. He can then check that out and continue working.

Managing inter-author cross-references

For an author to be able to insert a cross-reference in his document pointing into a document (file) being authored by somebody else, an appropriate destination point needs to exist in the destination file. The problems posed by this situation are:

- a. Cross-references imply links between files, thus filenames and locations must be both known and stable.
- b. There may not be a suitable destination point in the target file. Thus, the owner of the source file must be able to specify one and the owner of the destination file must be able to create it.

Probably the simplest solution is to have a convention by which authors can mark cross-references in their documents, to be established later by the editor at book compile time. Thus, as an example, inserting text such as “XREF *description*” in the document itself may be enough.

Another possible solution is to control the destination labels for cross-references manually, using *Cross-Ref* markers (exactly the same way as is done in LaTeX [7]). This will allow you to insert cross-references that point to a copy of the destination file. When this file is replaced with the master (e.g. when the editor compiles the book) the cross-reference will still work (as the same destination label would also have been inserted into the master destination file).

A.2 Changing the page layouts for different page sizes

Page sizes are changed by importing page layout formats from the format files in the *lib* subdirectory of the distribution.

Page layouts for the cover, the index, and all other files...

Each book template uses three different types of page layout definition sets—one for the cover, one for the index and one for everything else, i.e. the outline, toc, frontmatter, chapter, bibliography and appendix. Thus a predefined set of page layouts necessitate three different page layout format files.

Predefined page layouts

Table A.1 provides the complete list of predefined sets of page layouts, and the names of the three files in each set.

Table A.1 Sets of predefined page layouts to choose from.

	Page layouts	Paper size	Cutting?	Filesets for the Master Page formats
Normal	210 x 297	A4	No	normal/frontmatter/cover.fm normal/chapter/chapter.fm normal/usersguideIX.fm
	210 x 254	A4	Yes, at 254 mm	lib/MasterPages210x254Cover.fm lib/NormalMasterPages210x254.fm lib/MasterPages210x254Index.fm
	210 x 210	A4	Yes, at 210 mm	lib/MasterPages210x210Cover.fm lib/NormalMasterPages210x210.fm lib/MasterPages210x210Index.fm
	210 x 279.4	A4 width x US Letter height	No	lib/MasterPages210x279Cover.fm lib/NormalMasterPages210x279.fm lib/MasterPages210x279Index.fm
STOP	210 x 297	A4	No	STOP/frontmatter/cover.fm STOP/chapter/chapter.fm STOP/usersguideIX.fm
	210 x 254	A4	Yes, at 254 mm	lib/MasterPages210x254Cover.fm lib/STOPMasterPages210x254.fm lib/MasterPages210x254Index.fm
	210 x 210	A4	Yes, at 210 mm	lib/MasterPages210x210Cover.fm lib/STOPMasterPages210x210.fm lib/MasterPages210x210Index.fm
	210 x 279.4	A4 width x US Letter height	No	lib/MasterPages210x279Cover.fm lib/STOPMasterPages210x279.fm lib/MasterPages210x279Index.fm

Changing the page size

Changing the page size should be treated as a template customization and thus handled as discussed in Section 1.3, "Managing template customizations". This means that you should first change the page size of your reference template. You can then create new chapters using the template files in your reference template.

As an example let us suppose that you want to change the page size of your reference template from the default of A4 to that of 210 x 210 mm. The step-by-step procedure is as follows:

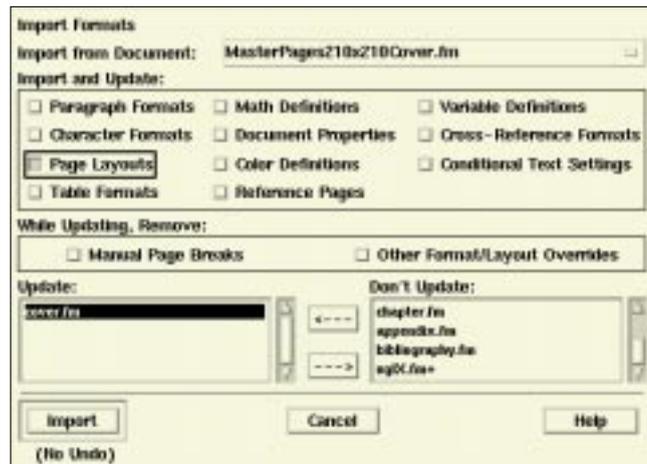
Preliminary set-up

1. If you have not yet done so, set-up your reference template as described in Section 1.2, "File organization and set-up".
2. Open the FrameMaker book file for your reference template with FrameMaker.



The cover page file

3. Open the format file for the new cover page layouts (*lib/MasterPages210x210Cover.fm*).
4. From the book file window of your reference template select *File:Import:Formats...* to pop up the *Import Formats* dialog box. Set it up as follows:
 - a. Make sure that in the *Import from Document:* field the file *MasterPages210x210Cover.fm* is selected.
 - b. De-select all format options except for *Page Layouts*.
 - c. Double click on all files under *Update:* except for the file *cover.fm* to move them to *Don't Update:*.
 - d. Click on *Import*.



The index file

5. Open the format file for the index page layouts (*lib/MasterPages210x210Cover.fm*)
6. Open the index file (*usersguideIX.fm*) in the book*.
7. From the book file window of your reference template select *File:Import:Formats...* to pop up the *Import Formats* dialog box. Set it up as follows:
 - a. Make sure that in the *Import from Document:* field the file *MasterPages210x210Index.fm* is selected.
 - b. Make sure that all format options except for *Page Layouts* are de-selected.
 - c. Make sure that the only file under *Update:* is the index file, i.e. *usersguideIX.fm*.
 - d. Click on *Import*.

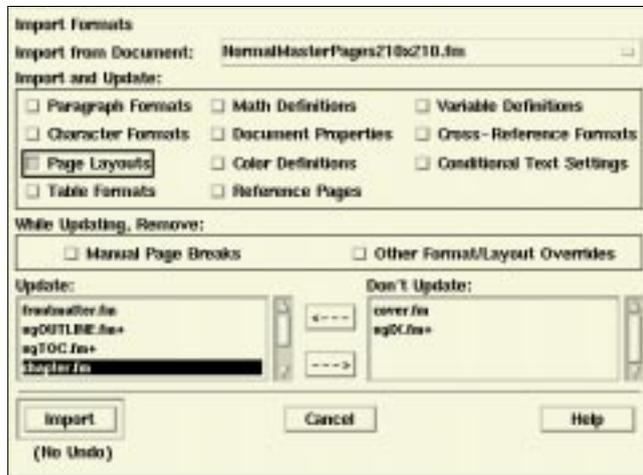
The other files

8. Open the format file (*lib/NormalMasterPages210x210.fm*) for the page layouts for all other files, i.e the outline, table of contents, frontmatter, chapter, bibliography and appendix.
9. Open all the automatically updated files[†], namely the outline (*usersguideOUTLINE.fm*) and the toc (*usersguideTOC.fm*).
10. From the reference template book file window select *File:Import:Formats...* to pop up the *Import Formats* dialog box. Set it up as follows:
 - a. Make sure that in the *Import from Document:* field the file *NormalMasterPages210x210.fm* is selected.

* To workaround a FrameMaker bug with the update of files locked for hypertext.

† For same reason as previous.

- b. Make sure that all format options except for *Page Layouts* are de-selected.
- c. Make sure that all files are under *Update*: except for the files *cover.fm* and *usersguideIX.fm*.
- d. Click on *Import*.



Finishing

11. Re-set the page numbering of the frontmatter to roman:
 - a. From the book window double-click the file *frontmatter/frontmatter.fm*.
 - b. Select *Format:Document:Numbering...* to pop up the *Numbering Properties* dialog.
 - c. For *Page # Style*: select *roman (iv)*.
 - d. Click *Set*.

Repeat this for the outline and the toc files but, as these are locked hypertext files they need to be first unlocked and then relocked. This is done by pointing in the document window and typing the keyboard shortcut **Esc F1k**, both for unlocking and for re-locking.

12. Regenerate and update the reference template book:
 - a. Select *File:Generate/Update* from the book *File* menu.
 - b. Make sure that all the automatically updated files are under *Generate*.
 - c. Click on *Generate*.
13. Save all files in the book (**Esc fs**) by holding down **shift** key + clicking on the book *File* menu to select *Save All Files in Book*.
14. Close, without saving, the files:
 - a. *lib/MasterPages210x210Cover.fm*
 - b. *lib/MasterPages210x210Index.fm*
 - c. *lib/NormalMasterPages210x210.fm*

15. Close all files in the book (**ESC** - **FC**) by holding down **SHIFT** key + clicking on the book *File* menu to select *Close All Files in Book*.

The page layouts of your reference template are now ready and you can proceed as suggested in Chapter 1, “Getting started”.

A.3 Deleting and re-adding automatically updated files

Deleting an automatically updated file is easy. The problem is to re-add them when necessary as this requires the specification of things like the source paragraphs or other objects, filename suffixes, etc.

Deleting an automatically updated file

1. From the book *File* menu, select *Rearrange Files...*
2. Select the file to be removed from the book and click on *Delete*.
3. Click on *Done*.



Note

This does not physically remove any files from the disk but simply modifies the book file.

Re-adding an automatically updated file

Each automatically updated file is different is a different case and needs to be treated individually. Table A.2 lists the exact information necessary to re-add each individual automatically updated file. The general procedure to re-add each of these files is as follows:

1. From the book *File* menu select *Add File...*
2. Select the appropriate type of *Generated List:* or *Generated Index:* (from Table A.2).
3. Select the component file to *Add File Before:* or to *Add File After:*.
4. Click on *Add*.
5. Specify the *Filename Suffix:* (from Table A.2).
6. Make sure that all the source names specified in Table A.2 are under the *Include...* column.
7. Make sure that *Create Hypertext Links* is selected.
8. Set *Starting Page Side:* to *Right* and *Page and Paragraph Numbering* to *Continue*.
9. Click *Set*.

Next time you regenerate and update the book FrameMaker will look for a file in the same location as the book file with the same name as the book plus the specified filename suffix.

Table A.2 Specifications of the automatically updated files.

Generated List Type	Filename Suffix	Source type	Source names
List of Paragraphs	OUTLINE	Paragraphs	ChapterNoNumber ChapterNumber ChapterNumberAppendix ChapterNumberContents ChapterNumberIndex ChapterTitle ChapterTitleContents ChapterTitleIndex
Table of Contents	TOC	Paragraphs	ChapterNoNumber ChapterNumber ChapterNumberAppendix ChapterNumberIndex ChapterNumberOutline ChapterSection ChapterSubSection ChapterSubSubSection ChapterTitle ChapterTitleIndex ChapterTitleOutline
List of Figures	LOF	Paragraphs	FigureCaption
List of Tables	LOT	Paragraphs	TableCaption
List of Listings	LOL	Paragraphs	ListingCaption
Standard Index	IX	Markers of Type	Index

A.4 Updating the formats of your document files

If you are in the authoring phase and you find you need to add or modify formats in a consistent manner throughout the files of your FrameMaker book, you will need to first customize your reference template, as discussed in Section 1.3, "Managing template customizations", and then update the formats of all your document files using the formats in your reference template. This section describes the mechanical procedure performing this update.

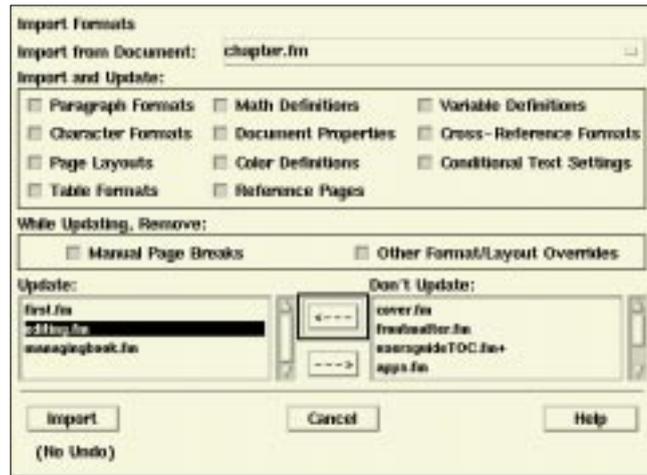
File types

Each component file in the book template defines a file type. When updating your document files from the template it is necessary to update each file from the template file of the same type.

Updating each file

1. Open the book files of both your reference template and your actual document.
2. Open all files in the book of your reference template (**Esc** **F0**, or **Shift** key + click on the book *File* menu to select *Open All Files in Book*).
3. For each component file* in the reference template (i.e. for each file type) update the document file of the same type, doing:
 - a. From the *File* menu of your document book window, select *Import:Formats...*
 - b. Under *Import from Document*: select a component file from the reference template, e.g. *chapter.fm*.
 - c. Make sure that all formats under *Import and Update*: are selected.
 - d. Remove overrides by selecting both items under *While Updating, Remove*:
 - e. Make sure that under *Update*: there are only the component files in your document of the same file type as that selected under *Import from Document*., e.g. all the chapter files of your document.
 - f. Click on *Import*.

* Due to a FrameMaker bug, each automatically updated file must be open during updating. Thus, before updating the outline, the toc and the index, make sure that they are open. Do not close them until the very end.



4. Regenerate and update your book document as explained in Section 3.1, "Routine book procedures".
5. Save all files in the book of your document (**Esc** **fS**, or **Shift** key + click on the book *File* menu to select *Save All Files in Book*).
6. Close all files in both books, i.e. both your document and your reference template (**Esc** **fS**, or **Shift** key + click on the book *File* menu to select *Close All Files in Book*).

A.5 Copying and creating *Zany* paragraphs

To use a *Zany* paragraph (see Section 2.8, "The zany paragraph formats") you need to create the format or import it from somewhere.

Copying *Zany* paragraph formats

This should be considered as a template customization and so should be treated as explained in Section 1.3, "Managing template customizations". The best file to add them to first is probably *chapter.fm* in your reference template, as follows:

1. Open the file *chapter.fm* of your reference template.
2. Open the file *Zany.fm* (in the *lib* directory of the distribution).
3. From the *File* menu of your chapter template file, select *Import:Formats...*
4. Under *Import from Document*: select *Zany.fm*.
5. Under *Import and Update*: select only:
 - *Paragraph Formats*
 - *Table Formats*
 - *Reference Pages*
6. Click on *Import*.
7. In the chapter template file, select *View:Reference Pages*.
8. Page down until you get to the *ZanyIcons* reference page.
9. Each frame adds a little less than a kilobyte to the filesize. Identify which ones you would like to keep and which ones you would rather not (remember that you should not use too many different ones in any one document anyway). For each one you want to throw away, do:
 - a. Select the frame and take note of its name (select *Graphics:Object Properties...*)
 - b. Delete the frame by hitting **Backspace**.
 - c. Pop up the *Paragraph Catalog* and click on *Delete...* at the bottom of it.
 - d. Delete the paragraph format with the same name as the frame you have just deleted.
10. Repeat this procedure for the frontmatter and appendix templates if you would also like to use zany paragraphs in those file types.

Creating your own Zany paragraph formats

The few predefined zany formats are intended mostly as an illustration of the use of these special iconed paragraphs. You are encouraged to create your own consistent set of such formats for your document sets. This helps to improve the consistency of your document set—as well as adding some identity.

All you need is an icon per zany format. The *lib* directory in the distribution contains an *icons* subdirectory with more than 50 icons that you can use. Once you have made or selected an icon [6], you can make a new zany format form it as follows:

Creating a new Zany frame

1. In your chapter template file, go to the *ZanyIcons* reference page.
2. Create a new frame with the graphics tool and give it an name of the form *ZanyIcon* where *Icon* is the name of the associated icon.
3. Using *Graphics:Properties...* set the width to 17.0 cm, the height to 1.0 cm and offset from left to 4.0 cm. Place it vertically on the page such that it does not overlap any other frame. Make sure that fill and pen are set to *None*.
4. Copy from any other frame the line and label grouped graphic object.
5. Select the new frame and paste (position of pasted object should be Top = 0.3 cm, Left = 3.4 cm).
6. Select the frame and import your prepared icon by saying *File:Import File*. Make sure you use *Copy into Document*, and click on *Import*.
7. In the *Import Graphic Scaling* dialog, specify an initial scaling of 100 dpi scaling and click on *Set*. Scale the icon further, as necessary, and move it (using **Control** + arrow keys, or equivalent) just to the left of the guiding vertical line.
8. Replace the label text with something more appropriate.

Creating a new Zany paragraph

9. Go to the *Body Pages* and create an instance of any of the previously defined zany paragraphs.
10. With the insertion point in this new paragraph, bring up the *Paragraph Designer (Format:Paragraphs:Designer...)*. Create the new paragraph by:
 - a. Change the paragraph tag name to the name you want to call your new zany paragraph, and click on *Apply*. In the *New Format* dialog, make sure that *Store in Catalog* and *Apply to Selection* are both selected, and click on *Create*.
 - b. Under *Properties*: select *Advanced*. Change *Frame Above* to use your newly created zany frame, and click on *Update All*.

A.6 Splitting a chapter into several files

If a chapter gets so big that working on it as a single file becomes too unwieldy, or if different parts of a chapter are being authored by different people, then it is useful to split up the chapter file into two or more files. Doing so however breaks the automatically set running header in all files following the first. This is because there is no occurrence of the source chapter title paragraph in these files.

Splitting the content

Make one or more copies of the chapter file that you want to split. Add the new copies to the book, using *File:Add File...* Open each file in turn and select and cut away all the pieces of material that should be contained in the other files.

Setting the running header in subsequent files

At the beginning of the second file and each one thereafter, insert a marker of type *Header/Footer \$1* with the exact text of the chapter title. This sets the running header on all left pages.

If a part file of a chapter contains no *ChapterSection* paragraphs, then you will also need to set the running header of all right pages. This is done in a similar way except that you would use a marker of type *Header/Footer \$2* instead.

Bibliography

- 1 **The Art of Indexing**
Larry S. Bonura; John Wiley & Sons, 1994.
- 2 **Writing Better Computer User Documentation: From Paper to Hypertext**
R. John Brockmann; John Wiley & Sons, 1990.
- 3 **Everything You Ever Wanted to Know about Documentation
(and should have asked)**
C. Fidge and D. Heagerty, Telecom Australia, Research Laboratories, July 1986.
- 4 **Adobe FrameMaker 5.5 User Guide**
Adobe Systems Incorporated, 1997.
- 5 **IXgen: Power Indexing & Marker Management Tool for FrameMaker**
Frank Stearns Associates, October 1996.
- 6 **The Icon Book: Visual Symbols for Computer Systems and Documentation**
William Horton; John Wiley & Sons, 1994.
- 7 **LaTeX, A Document Preparation System**
L. Lamport, Addison-Wesley, 2nd edition, Reading 1994.
- 8 **Writing User Documentation: A Practical Guide for those who want to be read.**
Roderick Low with Harriet Ford, John Pallot & James Hall, Prentice Hall, 1994.
- 9 **Typography: how to make it most legible**
Rolf F. Rehe; Design Research International, Fifth Revised Edition, 1984.
- 10 **The TypEncyclopedia: A User's Guide to Better Typography**
Frank J. Romano; R. R. Bowker Company, New York and London, 1984.

Numerics

2Col, 22
2-page spreads, 3, 26
3Col, 22
4Col, 22
5Col, 22

A

A4 paper size, 8, 49
acknowledgements, x
adding
 automatically updated files, 12, 54
 chapters, 12
 content files to book, 12, 13
advantages
 of good documentation, vi
 of importing by copy, 14
 of importing by reference, 14
appendices, 36
automatically updated files
 deleting, 54
 re-adding, 12, 54
availability, of templates, ix

B

background text, 43
BackgroundText, 43
base directory, 4
BiblioBody, 36
bibliographic references, 36
bibliography, 36, 61
BibliolItem, 36
BiblioSection, 36
Body, 18, 20
BodyPlaceholder, 20, 22, 24, 25
BodyTabular, 20
BodyVerbatim, 20

book

closing all files in, 57
deleting files in, 13
opening all files in, 43, 56
routine procedures, 42
saving all files in, 43, 57
updating, 42
book template, 1, 5
brain, left and right side, 2
breaking up information, 2

C

cerebral hemispheres, 2
chapter.fm, 6, 49, 58
ChapterNumber, 18
chapters, viii
 adding and deleting, 12
 section headings, 18
 splitting into files, 60
 title pages, 18
ChapterSection, 18, 60
ChapterSubSection, 18
ChapterSubSubSection, 18
ChapterTitle, 18
character formats, 20
 customizing, 20
 Input, 6, 20
 Output, 6, 20
checklist, for customizations, 6
ChepterSectionList, 18
chunks, manageable, 2
closing all files in book, 57
comments, sending, ix
component files, 4
 see also file types
 see also files
ConceptDescription, 20
ConceptTerm, 20
content, editing, 17

- conventions
 - in this manual, viii
 - naming new format, 6
- converting to WWW, 44
- cover, 34, 50
 - page layouts for, 49
- cover.fm*, 34, 49, 52
- cross-reference formats, 20
 - Bibliography*, 36
- cross-references
 - managing across authors, 48
- customizations, standard, 6
- customizing, 8
 - adding character tags, 20
 - adding paragraph tags, 30
 - and management, 6
 - checklist for, 6
 - naming new formats, 6
 - reference template for, 4, 6, 12
 - WWW conversion, 46
 - Zany* formats, 32
- cutting paper copies, 42, 49

D

- deleting
 - automatically updated files, 54
 - chapters, 12
 - content files from book, 13
- delivering the document, 41
- directories
 - icons*, 59
 - lib*, 8, 49, 50, 51, 52, 58
 - normal*, 49
 - STOP*, 49
- DL1Term*, 20
- docsys@ptsun00.cern.ch*, ix
- document
 - consistency, 6
 - delivering, 41
 - directory structure, 4
 - hierarchical structure, 2, 45
 - organization, 2
 - presentation, v
 - setting up, 4
 - styles, 2
 - usability, vi
- documentation, vi
 - and document usability, vii
 - and manpower problem, vi
 - as part of software, vi
 - different styles for, 2
 - for FrameMaker, viii
 - planning, vi
 - production difficulty, vii
- DocumentTitle*, 43
- double page spreads, 3, 26

E

- editing
 - content, 17
 - front matter, 34
 - multi-author, 48
 - running text, 20

F

- FcnCall*, 6, 30
- FcnInput*, 6, 30
- FcnOutput*, 6, 30
- FcnParameters*, 6, 30
- FcnSynopsis*, 6, 30
- feedback, ix
- Figure*, 24
- FigureCaption*, 24
- figures
 - inserting, 24
 - list of, 55
- file types, 56
 - appendix, 36
 - bibliography, 36
 - chapter, 6, 12
 - cover, 34, 50
 - frontmatter, 34
 - index, 38, 51
 - table of contents, 10
 - updating page layouts in, 51
- files, 4, 51, 52
 - adding and deleting in book, 12
 - chapter.fm*, 6, 49, 58
 - cover.fm*, 34, 49, 52
 - deleting in book, 13
 - FrameMaker hypertext, 39, 51
 - frontmatter.fm*, 52
 - locking and unlocking, 52
 - MasterPages210x210Cover.fm*, 49, 50, 51, 52
 - MasterPages210x210Index.fm*, 49
 - MasterPages210x254Cover.fm*, 49
 - MasterPages210x254Index.fm*, 49
 - MasterPages210x279Cover.fm*, 49
 - MasterPages210x279Index.fm*, 49
 - normal.tar*, 4
 - NormalMasterPages210x210.fm*, 49, 51, 52
 - NormalMasterPages210x254.fm*, 49
 - NormalMasterPages210x279.fm*, 49
 - organization of, 4
 - size increase per *Zany* format, 32
 - STOPMasterPages210x210.fm*, 49
 - STOPMasterPages210x254.fm*, 49
 - STOPMasterPages210x279.fm*, 49

- usersguideIX.fm*, 49, 51, 52
 - usersguideTOC.fm*, 51
 - Zany.fm*, 58
 - formats
 - character, *see* character formats
 - cross-reference, *see* cross-reference formats
 - graphics files, 14
 - naming new, 6
 - page layouts, *see* page layouts
 - paragraph, *see* paragraph formats
 - reference pages, *see* reference pages
 - table, *see* table formats
 - updating in document files, 56
 - variable, *see* variable formats
 - FrameMaker, viii
 - book template, 1, 5
 - documentation, viii
 - hypertext files, 39, 51
 - locking and unlocking, 52
 - Frank Stearns Associates, 38
 - front matter, 34
 - frontmatter.fm*, 52
 - function calls, 30
- ## G
- generated web, 45
 - getting started, 1
 - graphics
 - generating list of imported, 14
 - hiding, 14
 - imported file formats, 14
 - importing by copy, 14
 - importing by reference, 14
- ## H
- HardSpaces*, 28
 - Header/Footer \$1*, 60
 - Header/Footer \$2*, 60
 - headings, 18
 - hiding graphics, 14
 - hierarchy level indicator, 45
 - HTML, converting to, 44
 - hypertext files, 39, 51
 - in generated web, 45
 - locking and unlocking, 52
- ## I
- icons*, 59
 - Illustration*, 26
 - IllustrationCaption*, 26
 - IllustrationPlaceholder*, 26
 - IllustrationRight*, 26
 - illustrations, 26
 - on left pages, 26
 - on right pages, 26
 - imported files
 - generating a list of, 14
 - managing, 14
 - number of formats, 14
 - tips, 14
 - importing
 - graphics, by copy, 14
 - graphics, by reference, 14
 - listings, by copy, 28
 - listings, by reference, 28
 - indentation and spacing in listings, 28
 - index, 38, 51
 - building blocks, 38
 - importance of, 38
 - IXgen tool, 38
 - markers, 38
 - page layouts for, 49
 - information chunks, 2
 - Input*, 6, 20
 - inserting
 - figures, 24
 - multiple spaces, 28
 - tables, 22
 - inside cover, 34
 - installation, 4
 - introduction, 1
 - IXgen, 38
- ## K
- KeepRestOfPageClear*, 20
- ## L
- L1Alpha*, 20
 - L1Bullet*, 20
 - L1Continuation*, 20
 - L1Hyphen*, 20
 - L1Number*, 20
 - left brain, 2
 - lib*, 8, 49, 50, 51, 52, 58
 - linear logic, 2
 - list of
 - figures, 55
 - listings, 55
 - tables, 55
 - Listing*, 24
 - listing
 - list of, 55
 - ListingCaption*, 24
 - ListingLine*, 24, 25
 - ListingLineFirst*, 24, 28
 - listings, 24
 - importing by copy, 28
 - importing by reference, 28
 - indentation and spacing, 28
 - longer than 1 page, 24

ListingVerbatim, 24, 28
 logging template changes, 6
 logic, linear and sequential, 2

M

maintenance procedures, 47
 manageable chunks, 2
 managing

- imported files, 14
- inter-author cross-references, 48
- template customizations, 6

 markers, for the index, 38
MasterPages210x210Cover.fm, 49, 50, 51, 52
MasterPages210x210Index.fm, 49, 52
MasterPages210x254Cover.fm, 49
MasterPages210x254Index.fm, 49
MasterPages210x279Cover.fm, 49
MasterPages210x279Index.fm, 49
 multi-author editing, 6, 48

N

naming new formats, 6
 navigation aid, in generated web, 45
normal, 49
 normal style, 2
 normal template, 24

- figures, 24
- listings, 24
- snippets, 24

normal.tar, 4
NormalMasterPages210x210.fm, 49, 51, 52
NormalMasterPages210x254.fm, 49
NormalMasterPages210x279.fm, 49

O

opening all files in book, 43, 56
 organization

- of document files, 4
- of tasks into chapters, 10
- of the template, 4

 orientation aids, in generated web, 45
Output, 6, 20
 overview, 1

P

package, templates, ix
 page layouts

- 210mm x 210mm, 8
- 254mm x 210mm, 8
- changing, 49
- cover, index, other files, 49
- format filesets, 49

IllustrationRight, 26
 pre-defined, 49
Right, 26, 34
 selecting, 8
 updating different file types, 51
 page numbering, roman, 34
 page sizes, *see* page layouts
 paper copies, cutting, 42
 paper size

- A4, 8, 49
- A4 width x US Letter height, 49
- US Letter, 8

 paragraph formats, 20, 32
BiblioBody, 36
Bibliography, 61
BiblioItem, 36
BiblioSection, 36
Body, 18, 20
BodyPlaceHolder, 20, 22, 24, 25
BodyTabular, 20
BodyVerbatim, 20
ChapterNumber, 18
ChapterSection, 18, 60
ChapterSubSection, 18
ChapterSubSubSection, 18
ChapterTitle, 18
ChepterSectionList, 18
ConceptDescription, 20
ConceptTerm, 20
 customizing, 30
DL1Description, 20
DL1Term, *DL1Description*, 20
FcnCall, 6, 30
FcnInput, 6, 30
FcnOutput, 6, 30
FcnParameters, 6, 30
FcnSynopsis, 6, 30
FigureCaption, 24
IllustrationCaption, 26
IllustrationPlaceHolder, 26
KeepRestOfPageClear, 20
L1Alpha, 20
L1Bullet, 20
L1ContinuationVerbatim, 20
L1Hyphen, 20
L1Number, 20
ListingCaption, 24
ListingLine, 24, 25
ListingLineFirst, 24, 28
ListingVerbatim, 24, 28
Quote, 20
QuoteBullet, 20
SideHeadSection, 6, 30
SideHeadStep, 20
SideHeadSubSection, 6, 30
TableCaption, 22
Zany, *see* *Zany* formats

- part of software, documentation, vi
- PDF, 46
- planning, user guide, 10
- pre-defined formats
 - characters, *see* character formats
 - cross-reference, *see* cross-reference formats
 - page layouts, *see* page layouts
 - paragraphs, *see* paragraph formats
 - reference pages, *see* reference pages
 - tables, *see* table formats
 - variable, *see* variable formats
- presentation, v
 - and usability, vi
 - importance of, vi
- printing, to standard sized paper, 42
- procedures
 - book routine, 42
 - set-up and maintenance, 47
- process, 4, 6, 48

Q

- Quote*, 20
- QuoteBullet*, 20

R

- re-adding
 - automatically updated files, 12, 54
- reference pages
 - ZanyIcons*, 58, 59
- reference template, 4
- references, bibliographic, 36
- required software, ix
- Right*, 26, 34
- right brain, 2
- roman page numbering, 34
- routine book procedures, 42
- running footer, 43
- running header
 - in multi-file chapters, 60

S

- saving all files in book, 43, 57
- saving as PDF, 46
- section headings, 18
- sequential logic, 2
- setting up your document, 4
- set-up procedures, 47
- SideHeadSection*, 6, 30
- SideHeadStep*, 20
- SideHeadSubSection*, 6, 30
- Smart Spaces*, 28
- Snippet*, 24
- software, required, ix
- spaces, hard, 28

- spacing and indentation in listings, 28
- special sections, 30
- splitting a chapter into several files, 60
- spreads, 2-page, 3
- standard customizations, 6
- STOP*, 49
- STOP method, viii, 2
- STOP template
 - illustrations, 26
 - STOPMasterPages210x210.fm*, 49
 - STOPMasterPages210x254.fm*, 49
 - STOPMasterPages210x279.fm*, 49
- structure, 4
- structured report style, 2
- structured writing, 2
- style, 2
 - choosing, 2
 - normal, 2
 - of documentation, 2
 - STOP, 2
 - structured report, 2
- submissions of material to editor, 48
- subsection headings, 18
- support, ix

T

- table formats, 22
 - 2Col*, *3Col*, *4Col*, *5Col*, 22
 - Figure*, 24
 - Illustration*, 26
 - Listing*, 24
 - Snippet*, 24
 - ZanyTextSection*, 32
- table of contents, 10
- TableCaption*, 22
- tables
 - inserting, 22
 - list of, 55
 - tips, 22
- task-oriented approach, 10
- template customizations, 6, 8
 - managing, 6
 - page layouts, 49
- templates
 - and availability, ix
 - and efficiency, vi, vii
 - and results, vii
 - availability, ix
 - chapter title page, 18
 - customizing WWW conversion, 46
 - directory structure, 4
 - file types, *see* file types
 - for better results, vii
 - getting started, 1
 - logging changes, 6
 - normal.tar*, 4

- options, 1
- package, ix
- problem addressed by, vii
- required software, ix
- setting up, 4
- subdirectory, 4
- support, ix
- two different styles, 2

two-page spreads, 3, 26

U

- updating
 - formats, 56
 - FrameMaker book, 42
- US Letter paper size, 8
- usability
 - of documentation, vii
- user documentation, *see* documentation
- user guide templates, *see* templates
 - usersguideIX.fm*, 49, 51, 52
 - usersguideOUTLINE.fm*, 51
 - usersguideTOC.fm*, 51

V

- variable formats

- BackgroundText*, 43
- DocumentTitle*, 43
- Header/Footer \$1*, 60
- Header/Footer \$2*, 60
- propagating across book, 43
- resetting in a single file, 43
- running footer, 43

version control, 48

W

- web, generated, 45
- WWW, converting to, 44

Z

- zanni, 32
- Zany formats
 - and file size, 32
 - copying, 32, 58
 - creating, 32, 58, 59
 - number of, 32
- Zany* formats, 32
- Zany* sections, 32
- Zany.fm*, 58
- ZanyIcons*, 58, 59
- ZanyTextSection*, 32