

IRIS InSight™ Professional Publisher User's Guide

Document Number 007-2863-002

CONTRIBUTORS

Written by Jed Hartman, Debbie Myers, and Pam Sogard

Updated by Linda Rae Sande

Illustrated by Dany Galgani

Production by Linda Rae Sande

Engineering contributions by David Clarke, Greg Ferguson, Victor Riley, Steve Schmitt, and Lorrie Williams

St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower image courtesy of Xavier Berenguer, Animatica.

© Copyright 1992-1995, 1999, Silicon Graphics, Inc. — All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED AND RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in the Rights in Data clause at FAR 52.227-14 and/or in similar or successor clauses in the FAR, or in the DOD, DOE or NASA FAR Supplements. Unpublished rights reserved under the Copyright Laws of the United States.

Contractor/manufacturer is Silicon Graphics, Inc., 1600 Amphitheatre Pkwy., Mountain View, CA 94043-1351.

Silicon Graphics, IRIS, and IRIX are registered trademarks, and Indy, IndyCam, InPerson, IRIS InSight, and IRIS Inventor are trademarks, of Silicon Graphics, Inc. FrameMaker and PostScript are registered trademarks and Adobe Illustrator is a trademark (which may be registered in certain jurisdictions) of Adobe Systems Incorporated. DynaWeb is a trademark of Inso Corporation. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Additional License Restrictions: The Distribution and Use of IRIS InSight Professional Publisher are limited to the SGI platform. Use is limited to the SGI DTDs and for the sole purpose of electronic publication of information directly related to a product or service, available on or for the SGI platform.

IRIS InSight™ Professional Publisher User's Guide
Document Number 007-2863-002

What's New in This Guide

This update of IRIS InSight Professional Publisher User's Guide documents the 4.2 tools for bookbuilding. These tools include support for the new SGIDOCBK DTD.

New Features Documented

Bookbuild tools can be installed on systems running IRIX 6.2 and IRIX 6.5.X and include updated scripts, a new Makefile template, and a new set of error messages.

The online appearance of several character tags have changed in the SGIDOCBK DTD. The SGIDOC DTD is still supported in 4.2 tools, however.

Note: The functionality of the two DTDs is essentially the same; the online appearance is different. Before converting all your bookbuilding to the new SGIDOCBK DTD, you'll want to perform a sample bookbuild and view it in IRIS InSight to ensure its online appearance matches your company's style guide. If you need to build books for the SJIS Japanese locale or any Chinese or Korean languages and locales, you must use the SGIDOCBK DTD.

Major Documentation Changes

The following changes were made to this book:

Chapter 2, "Preparing Your Workstation," contains the following changes:

- Updated list of software subsystems and configuration requirements
- Installation instructions and distribution locations for software

Chapter 3, "Using the IPTemplates to Author Documents," is new.

Chapter 4, "Using FrameMaker Files and Template Features," (formerly Chapter 5) contains expanded instructions on how to create:

- Links within a book
- Index entries

and new instructions on how to create:

- Reference (man) page links
- Inline media links
- Links to URLs and applications

Chapter 5, "Working With Figures," (formerly Chapter 6 and Appendix B), contains the following updated lists:

- Supported figure formats
- Screen capture tools and their software subsystems

Chapter 6, "Building Online Books," (formerly Chapter 3), contains updated information about the *Makefile* and *make* commands for the new 4.2 build tools.

Chapter 7, "Finding and Fixing Online Build Errors," (formerly Chapter 4), contains an updated list of errors and their meanings.

Chapter 8, "Creating Online Help," (formerly Chapter 7), contains instructions on viewing and testing help using the updated SGIHelp graphical user interlace.

Appendix A, "IRIS Insight Bookbuilding Architecture," (formerly Appendix C) is an updated overview of the bookbuilding process.

Record of Revision

Version	Description
002	June 1999 Incorporates information for the release of <i>insight_dev</i> and 4.2 build tools with IRIX 6.5.5.

Contents

What's New in This Guide	iii
New Features Documented	iii
Major Documentation Changes	iii
Record of Revision	v
List of Figures	xv
List of Tables	xvii
About This Guide	xix
Audience for This Guide	xx
Organization of This Guide	xx
Supplementary Reading	xxi
1. Getting Started	1
Before You Begin	2
Viewer Features to Keep in Mind	3
Guidelines for Organizing an Online Book	4
"Chunk" the Text	4
Provide a Linked Chapter Summary	4
Write Descriptive, Standalone Headings	5
Use Terms Consistently	5
Avoid Print-Oriented Wording	6
Guidelines for Cross-Referencing in Online Books	6
Write Informative Cross-References	6
Cross-Reference Repeated Information	7
Use Cross-Book References Carefully	7
Overview of the Book Development Process	8

- 2. Preparing Your Workstation 9**
 - Summary of Software/Configuration Requirements 9
 - Setting Up Tools and Environment 10
 - Installing Subsystems 11
 - Installing Subsystems Using Inst 14
 - Installing Subsystems Using Software Manager 14
 - Setting Up a Workarea 15
 - Organizing a Local Working Directory 16
 - Standard Files in a Local Working Directory 17
 - Observing Naming Conventions 19
 - Copying Template Files to Your Working Directory 22
- 3. Using the IP Templates to Author Documents 25**
 - IP Templates - A Quick Tutorial 26
 - IPBook 26
 - IPFront.doc 27
 - IPTOC.doc, LOE.doc, LOF.doc, LOT.doc, and IX.doc 27
 - IPIntro.doc 28
 - IPPartTab.doc and IPTab.doc 28
 - IPChap.doc and IPNoChap.doc 28
 - IPAppendix.doc 28
 - IPGlossary.doc 29
 - Applying the Templates to Existing Files 29
 - Paragraph Tag Naming Scheme 31
 - Chapter and Appendix Numbers and Titles 31
 - Headings 31
 - Second-Level Heading 32
 - Second-Level Heading 32

- Lists 32
 - Bulleted Lists 33
 - Square Bulleted Lists 33
 - Dashed Lists 34
 - Ordered Lists and Sequential Steps 34
 - Hanging Lists 35
 - Tabular Lists 36
- Text Paragraph Tags 37
- Code and Example Paragraph Tags 38
 - Example Paragraph Tags 38
 - Code Paragraph Tags 39
 - Numbered Code and Example Blocks 41
- Messages and Explanations 42
- Checkoff Paragraph Tags 43
- Equation Paragraph Tags 43
 - Guidelines for Typing Mathematical Equations 44
 - Mathematical Symbol Usage Guidelines for All Documents 45
- Figure Tags 46
- Tables 47
 - Customizing Tables 48
 - Inserting Figures in Tables and Tabular Lists 49
- Hint, Tip, Note, Caution, Warning, and Shortcut Paragraph Tags 50
 - Hint Paragraph Tags 51
 - Tip Paragraph Tags 52
 - Note Paragraph Tags 53
 - Caution Paragraph Tags 54
 - Warning Paragraph Tags 55
 - Adding the Warning Icon to Warning Paragraph Tags 56
 - Shortcut Paragraph Tags 57
- Flag Paragraph Tags 57
 - Advantages of Using the Flag Tags 58
 - Adding the Appropriate Icon to a Flag Paragraph 58
 - Managing Flag Paragraphs 59

- Footnote Paragraph Tag 60
- Page Break Paragraph Tag 60
- Character Tags 60
- Cross References 62
- Markers 63
- Conditional Text 64
- InSight Inline Object Tags 64
- Help Paragraph Tags 64
- Creating a Book File 65
 - Added Versus Generated Files 65
 - Adding Text Files to Your Book File 66
 - Adding Generated Files to Your Book File 69
 - Rearranging and Checking the Setup for Your Files 70
 - Generating the TOC, LOE, LOF, LOT and Index 70
 - Troubleshooting Autonumbering Problems 71
- 4. Using FrameMaker Files and Template Features 73**
 - Tagging FrameMaker Files 73
 - Using Character Tags, Not Character Formats 74
 - Using Valid Tag Sequences 74
 - Using the Correct Character Tag 74
 - Creating Equations as Figures 74
 - Using Character Tags on Tabs, Soft Returns, or Hard Returns 75
 - Using Tabs in Code 75
 - Creating Valid Copyright and Trademark Symbols 76
 - Creating Valid Dashes 76
 - Creating Footnotes 76

- Creating Online Links 77
 - About Automatic Links 77
 - About Writer-Generated Links 77
 - Creating Cross-Reference Links Within a Book 79
 - Creating Cross-Reference Links to Other Books 82
 - Creating Glossary Links 84
 - Creating Index Entries 84
 - Creating Reference (Man) Page Links 89
 - Creating Inline Media Links 90
 - Creating Links to URLs and Applications 90
- 5. **Working With Figures** 93
 - Supported Image Formats 94
 - Image File Naming Conventions 94
 - The Figure Subdirectories 95
 - Adding Image Files to the Makefile 96
 - About FrameMaker Art 98
 - Capturing Images 99
 - Tools and Where to Get Them 99
 - Screen Capture File Names 100
 - Screen Capture Procedure 100
 - Processing and Importing Image Files 103
 - Using Unsupported Image Types 104
 - Using Custom *print* Files 105
 - Things to Remember About Figures 106
- 6. **Building Online Books** 109
 - Editing the Makefile 110
 - Files and Directories Generated by *make* 113
 - Intermediate Files 114
 - Error Files 115
 - The *books* Directory 115
 - Arguments to the *make* Command 116

- Using *make* Commands 118
 - Initial *make*: *make book.full* 118
 - Debugging Individual Files: *filename.err* 119
 - Making a Viewable Book: *make book* 120
 - Verifying an Error-Free Book: *make book.full* 120
- If *make* Doesn't Work 121
- make book* Error Messages 122
 - Problem: File Already Locked 122
 - Problem: Can't Open File 122
 - Problem: *commonbookdefs* File Errors 123
 - Problem: Can't Open *booklist.txt* File 123
 - Problem: Must Have *impr_rip.sw.impr* 123
 - Problem: Cannot Map *libnsl.so* Library 123
- Viewing a Built Book 124
 - Viewing an Uninstalled Book 124
 - Viewing an Installed Book 124
- 7. Finding and Fixing Online Build Errors 127**
 - About Error Messages 128
 - Error Files 128
 - About SGML 128
 - Example SGML File 129
 - SGML Tag Structure 133
 - Checking the *.full* File 133
 - An Example *.full* File 134
 - The Translation Error and Warning Report 135
 - The Internal SGI Doc Report 136
 - The Link QA Report 137
 - Glossary QA Report 141
 - Finding and Fixing Errors 142
 - Types of Errors 142
 - Finding the Errors in FrameMaker Files 142
 - MIF-to-SGML Translation Errors 143
 - Unknown Errors 158

	Checking for Display Problems	159
8.	Creating Online Help	161
	Overview of Help	161
	Application	162
	Helpmap	162
	Document	163
	Tagging FrameMaker Files for Help	163
	Help Buttons Versus Help Menus and Context Sensitive Help	164
	The Helpmap File	165
	An Explanation of the Helpmap File	166
	A Sample Helpmap File	167
	Creating A Rollover Menu	168
	Creating Context Sensitive Help	168
	Setting Up the Fallback Mechanism for Context Sensitive Help	169
	Using the World Wide Web from Your Help Menu	170
	Testing the Online Help	170
	Testing Help From the Application	171
	Testing Help Without the Application	173
	Telling Users Which Subsystems to Install	174
	Troubleshooting	174
	No Help Available	175
	The Wrong Help Topic Appears	175
	Missing Items on the Help Menu	176
A.	IRIS Insight Bookbuilding Architecture	177
	Bookbuilding Architecture	177
	Glossary	179
	Index	183

List of Figures

Figure 2-1	A Typical SGI Workarea	16
Figure 2-2	Files and Directories in a Working Directory	17
Figure 3-1	Dummy Figure Title Using FigTitle Paragraph Tag	46
Figure 4-1	Diagram of Cross Reference Targets	80
Figure 6-1	Files Produced During the make Process	114
Figure 7-1	FrameMaker Example File	130
Figure 7-2	SGML Opening and Closing Tags	133
Figure 7-3	Figure Without a Cross Reference Target Marker	137
Figure 7-4	Figure with a Cross Reference Target Marker	138
Figure 8-1	Standard Flow for Help	162
Figure 8-2	A Sample Help Menu	172
Figure 8-3	A Sample SGI Help Graphical User Interface	172
Figure A-1	Architecture for Book-Building	177

List of Tables

Table 1-1	Suggestions for Avoiding Print-Oriented Wording	6
Table 2-1	Subsystems to Install on IRIX 6.2 Systems	11
Table 2-2	Subsystems to Install on IRIX 6.5 Systems	12
Table 2-3	Files and Directories in a Book Directory	18
Table 2-4	Guidelines for Determining Makefile and File Suffixes	20
Table 3-1	Creating Mathematical Symbols	45
Table 3-2	Real $N \times N$ Operations (TableTitle)	50
Table 3-3	Anchored Frame Specs for Warning Tags	56
Table 3-4	Anchored Frame Specs for Flag Tags	59
Table 3-5	Character Formats	61
Table 3-6	Page Setup for Text Files (No Part Tabs or Chapter Tabs)	66
Table 3-7	Page Setup for Text Files (With Chapter Tabs)	67
Table 3-8	Page Setup for Text Files (With Part Tabs)	67
Table 3-9	Page Setup for Text Files (With Part Tabs and Chapter Tabs)	68
Table 3-10	Page Setup for Generated Files	69
Table 4-1	Cross-Reference Link Types	78
Table 4-2	Cross-Reference Format by Paragraph Tag	81
Table 5-1	Image Formats Supported by the Build Tools	94
Table 5-2	Image File Naming Conventions	95
Table 5-3	Image Formats for Hard and Soft Copy	106
Table 6-1	FrameMaker Files to Include and Omit in Makefile	112
Table 6-2	<i>make</i> Commands For the Entire Book	116
Table 6-3	<i>make</i> Commands for Single FrameMaker Files	116
Table 6-4	<i>make</i> Commands for Figures	117
Table 6-5	<i>make</i> Commands for Removing Files	117
Table 7-1	Cross Reference Target Scenarios	140

About This Guide

This guide explains how to prepare and produce technical documentation using the IRIS InSight Professional Publisher templates and online bookbuilding tools. Users can read online manuals using the IRIS InSight document viewer, proprietary software that exploits the graphics capabilities of Silicon Graphics workstations. IRIS InSight allows readers to launch manual pages as well as applications from within an online book.

IRIS InfoSearch, a Web-based search utility, is available to view all online documentation on systems running IRIX 6.3 and above.

Online “help” is intended to be viewed through SGI Help, an application derived from IRIS InSight. SGI Help presents discrete pieces of information about an application from within the application, as opposed to the book-oriented approach of IRIS InSight. Like IRIS InSight, however, SGI Help allows readers to view portions of technical manuals and to launch other applications.

To create and edit manuals and help material to be viewed with IRIS InSight, InfoSearch, or SGI Help, use the FrameMaker document publishing application from Adobe Systems Incorporated. Use the FrameMaker tags and templates provided as part of IRIS InSight Professional Publisher Templates (IPTemplates) to structure document files so that they can be converted to Standard Generalized Markup Language (SGML) for online viewing. For manuals delivered in both print and online versions, both versions are derived from a single FrameMaker source file.

The InSight-viewable version of a technical manual is delivered as an installable software image, usually a subsystem within a software product. For this reason, you must put the component files of a viewable book through a special preparation process referred to as a *bookbuild*. The bookbuilding process translates FrameMaker source files to SGML using the *SGIDOCBK DTD* or *SGIDOC DTD*, and generates the additional files that are required for the installable software image.

Audience for This Guide

The audience for this guide includes writers who are responsible for revising existing technical manuals and creating new ones, and production editors who prepare a writer's document source files for delivery to print vendors or to a software release group. (It's assumed that a *production editor* is associated with the book and knows something about your company's document-production issues. If you don't have a production editor, you may have to handle production tasks yourself.) Experienced writers and production editors who are familiar with the SGI's book development process can refer to this guide for selected procedures and points of information. New writers and production editors should rely on this guide to learn the tools and processes with which to develop online manuals.

Readers of this guide are assumed to be experienced FrameMaker users, although detailed instructions on how to use FrameMaker's features are included.

Organization of This Guide

This guide is organized to present the book development procedures in an unbroken sequence.

- Chapter 1, "Getting Started," recommends ways to become familiar with the IRIS InSight viewer and to adjust your writing style for the online reader.
- Chapter 2, "Preparing Your Workstation," explains how to organize the area of your workstation that is used during book development and how to set up the software that you will use.
- Chapter 3, "Using the IPTemplates to Author Documents," describes the templates and tags that make up the IPTemplates and how to use them when creating your documents. It also explains how to create and set up a FrameMaker bookfile.
- Chapter 4, "Using FrameMaker Files and Template Features," explains how to use the SGI document templates and their associated tags in creating a FrameMaker source file. It also explains file naming conventions and the specifics of applying FrameMaker tags to ensure the successful build of an online manual.
- Chapter 5, "Working With Figures," identifies acceptable formats for figure files and explains the purpose of directories where figure files are stored. It also explains how to process supported and unsupported figure types so that they are viewable in an online manual.

- Chapter 6, “Building Online Books,” contains the procedures for converting FrameMaker source files into an online manual. It also explains how to view an online book created by the bookbuilding process.
- Chapter 7, “Finding and Fixing Online Build Errors,” explains how to locate and correct problems that occur in the bookbuilding process.
- Chapter 8, “Creating Online Help” explains how to create online help for the SGI Help viewer. Help information can be either a separate document or part of an online manual.

In addition, this guide contains the following appendix:

- Appendix A, “IRIS Insight Bookbuilding Architecture,” diagrams the bookbuilding architecture and process.

Supplementary Reading

Refer to these documents to supplement the information in this guide:

- *Designing and Writing Online Documentation: Help Files to Hypertext*, by William K. Horton. Published by John Wiley and Sons, Inc. For writers who want to know more about writing for the online medium.
- *Illustrating Computer Documentation: The Art of Presenting Information Graphically on Paper and Online*, by William Horton. Published by John Wiley and Sons, Inc. Provides guidelines for illustrating and designing effective graphics in either the paper or online medium.
- *SGI Editorial Style Guide*. Published by the SGI editorial staff to describe SGI editorial standards and preferences.
- *IPTemplate*. Published by SGI as part of the InSight Professional Publisher distribution. These are the templates used to produce manuals. The *IPchap.doc* file in the `/usr/share/Insight/templates/frame/IPTemplate` directory contains information on how to use the various tags and features of the templates.
- *Using FrameMaker and FrameMaker Reference*. Published by Adobe. These books document basic and advanced FrameMaker features.

Getting Started

This chapter explains what to do to prepare for the book development process. The chapter recommends ways to learn about the online medium and how to adjust your writing style for the online viewing audience. It also gives an overview of the book development process and tells you where to look in this book for an explanation of each step in the process.

This chapter contains these sections:

- “Before You Begin”
- “Viewer Features to Keep in Mind”
- “Guidelines for Organizing an Online Book”
- “Guidelines for Cross-Referencing in Online Books”
- “Overview of the Book Development Process”

Note: The writing guidelines in this chapter are very brief. For a more thorough discussion of effective online writing style, refer to the books listed as “Supplementary Reading” on page xxi.

Before You Begin

Before you write or revise a book for online presentation, take a look at an existing online book. Start by launching IRIS InSight if it's not already running; then try some of these features:

1. Select a topic from the Help menu on the InSight main window's menu bar.
2. Open a book by double-clicking its title.
3. Click some hypertext elements:
 - blue text (a link to a location in the current book)
 - red text (a link to a location in a different book, or to launch an application)
 - blue italicized word or phrase (a glossary item)
 - rectangle containing an arrow pointing straight right (a margin note or a footnote)
4. Navigate through the book. Try using the Table of Contents, the List of Examples, the List of Figures, and the List of Tables (choose Book > View) and the Index (choose Book > Index).
5. Click a figure to bring up a restricted view of the image.
6. Click a table title to bring up a restricted view of the table.
7. Use the text search feature.
8. Read through an entire chapter to get a feel for the flow of online text.

If you are using a system running the IRIX 6.3 or higher operating system, launch InfoSearch to view an online book through a browser.

1. Choose Toolchest > Help > Info Search to launch the InfoSearch utility.
2. Click *Online Books* to display the bookshelves installed on your system.
3. Click one bookshelf name and then click on any book's title.
4. Click one of the chapter's titles.
5. Click some hypertext elements:
 - blue text (a link to a location in the current book or to another book)
 - navigation arrows (located at the top and bottom of each major section)
6. Navigate through the book. Try using the navigation arrows.

7. Click on the word *InfoSearch* at the top of the page to return to the main InfoSearch page.
8. Use the text search feature.

Encourage engineering reviewers to review both the soft copy and the hard copy of your book. This is particularly important if your book is not shipped in hard copy by default.

Viewer Features to Keep in Mind

When writing an online book, it might help you to be aware of the IRIS InSight features that are available to your readers. Readers can

- search for specific text within one book or across an entire library
- navigate through a book by clicking entries in the Table of Contents, List of Example, List of Figures, List of Tables, or Index, or by clicking cross-references
- change font sizes in book windows
- open several books at the same time
- retrace their steps within a session
- get help by way of the Help menu
- print sections, chapters, or entire books
- launch applications by clicking bold red text within InSight
- view images, movies, and IRIS Inventor models inline
- listen to audio files
- view images and tables either inline or in their own separate windows

In addition, SGI online books are available for viewing on the Web via DynaWeb at <http://techpubs.sgi.com/> or through InfoSearch (for systems running IRIX 6.3 and above). InfoSearch allows users to search and browse virtually all SGI documentation: online books, man pages and release notes. Although InfoSearch does not contain every feature offered by IRIS InSight, man, or relnotes subsystems, it enables your reader to find information contained in all these SGI-supported forms of documentation.

Guidelines for Organizing an Online Book

This section includes some tips for writing good online documentation. These tips are primarily aimed at preventing users from getting lost in the online version of a book. When writing your online book, remember that users might not see the pages in order. Further, they might not know exactly where they are in the book at any given time.

Here's an overview of basic online writing tips:

- Write in chunks. Try to make each section answer one question. Provide links to supplemental information.
- At the beginning of each chapter, provide a list of links to each of the chapter's major headings. See "Provide a Linked Chapter Summary" on page 4 for details.
- Write good section headings. Make them distinct and make them descriptive enough to stand alone. See "Write Descriptive, Standalone Headings" on page 5 for details.
- Be consistent in your use of terms.
- Avoid print-oriented wording. See "Avoid Print-Oriented Wording" on page 6.

"Chunk" the Text

Usability tests indicate that readers get lost after scrolling through more than three windowfuls of text without headings. Your book will be easier to read and navigate online if you organize text in *chunks*: brief, to-the-point, stand-alone sections. Think of the topics in your book as index cards scattered on a desk. Readers may jump to a topic without seeing what comes before or after it. Make each section answer one question, then provide links to supplemental information.

Provide a Linked Chapter Summary

At the beginning of each chapter, provide a bulleted list of all first-level headings in that chapter. Usability tests indicate that, when a chapter begins with a bulleted list, users expect the items in the list to be links. Use the FrameMaker cross-reference tool to create the list, so that each section heading will be a link in the online version of the book.

Of course, in some books it may not make sense to begin each chapter with a linked summary. As always, use your best judgment of what is appropriate for your book.

Write Descriptive, Standalone Headings

A good online heading tells the reader what is contained in the section without relying on information contained in the parent heading. For instance, if a section called “Comparing Files” contains a subsection that describes how to use *diff*, name the subsection, “Using *diff* to Compare Files,” rather than simply “*diff*.”

Since your readers may enter the section from a different chapter or even a different book, using “*diff*” by itself as a subsection title doesn’t provide enough context information; it requires the reader to skim the previous section and to know what the *diff* utility is used for.

To make it easy for a reader to stay “located” in your book, stick to one topic per section. If the subject changes, write a new heading. Don’t worry about having too many headings; a large number of headings will make your document more usable and won’t overload it.

Use Terms Consistently

Online readers leap quickly from section to section. While looking for information on a particular topic, a reader might consecutively read eight different paragraphs from various chapters in your book, and others from another book as well. So, for example, if you call something a “short name” in one chapter and a “short title” in another, and another writer calls it a “shorty” in a different book, readers may get confused. This kind of inconsistency is inadvisable, in a hard copy book, but it’s deadly online. If you change a term partway through the writing process, use FrameMaker’s Edit > Find/Change tool to make sure you change all the occurrences of the old term.

Avoid Print-Oriented Wording

Online readers approach information from many different directions. Don't assume that a reader knows or has read information that isn't currently on the screen. Avoid wording that refers to location within a book, unless you also create a link to that information. Table 1-1 gives examples of how to avoid print-oriented wording.

Table 1-1 Suggestions for Avoiding Print-Oriented Wording

Use...	Instead of...
"in the section named..."	"as shown above (or below)"
"in Chapter 3, you learned that..."	"by now you have learned that"
"in the previous section, <section title>,"	"in the previous section"

Guidelines for Cross-Referencing in Online Books

In the online version of your book, local cross-references appear in blue and are linked to the referenced material. This gives your readers another way to navigate the book. Well-designed cross-references get information to the people who need it, and keep it out of the way of those who don't. Poorly-designed cross-references can lose readers in annoying and confusing wild-goose chases. This section offers some guidelines for good cross-referencing.

Write Informative Cross-References

Include enough information in a cross-reference so a reader can decide whether to follow it. For example, "See the *Origin200 Owner's Guide*" doesn't tell the reader what information you are referencing. A better reference might be, "See the section called 'Setting Up Your System' in the *Origin200 Owner's Guide* for more information about plugging in the power cord." This reference tells the readers exactly what information they can expect to find at the other end of the link.

Cross-Reference Repeated Information

Avoid including the same information in multiple places in your book. Instead, put it in the most important place, then create cross-references to it as necessary. This way, readers who need the information can just click the cross-reference, while readers who don't need it won't need to scroll through it. And when you need to update the information on that topic, you won't have to remember to update it in more than one place.

Use Cross-Book References Carefully

Linked cross-references between different books are one of the most useful features of online documentation. They do, however, involve risks. Another writer could change the organization, the content, or even the name of the referenced book. Or the book might be on a different release schedule, which means your cross-reference might be destroyed when a new version of the referenced book is released. Here are some guidelines to minimize such risks:

- Avoid gratuitous cross-book references.
- Notify the authors of books that you plan to cross-reference. The writers can give you an idea of how far along their books are in the revision cycle and whether the content and organization are stable.
- When referring to documentation for an optional product, make it very clear that the document is optional and that the reader might not have it: "If you purchased the C++ option, see the *C++ User's Guide* for more information."
- The safest cross-references are to local books and books that ship with the same software product as your book.

Overview of the Book Development Process

The book development process comprises five general steps:

1. Prepare your workstation.

The first step in developing a book is to set up the software tools on your workstation. See Chapter 2, "Preparing Your Workstation."

2. Write your book or update an existing book.

Use the IPTemplates and follow the directions in Chapter 3, "Using the IPTemplates to Author Documents," and Chapter 4, "Using FrameMaker Files and Template Features."

3. Add figures.

Create anchored frames and import your figures as described in Chapter 5, "Working With Figures."

4. Build and debug the online version of your book.

Build the online version of the book periodically; see Chapter 6, "Building Online Books." If you encounter errors during the build process or while viewing the completed online book, see Chapter 7, "Finding and Fixing Online Build Errors."

5. Add help topics to your FrameMaker files and create helpmaps if some or all of your book's contents act as "help" for an application or utility.

See Chapter 8, "Creating Online Help," for instructions for making your book into an SGHelp document.

Preparing Your Workstation

The book development process requires access to the bookbuilding and viewing tools.

This chapter explains how to prepare your workstation for book development. The chapter contains these sections:

- “Summary of Software/Configuration Requirements”
- “Setting Up Tools and Environment”
- “Setting Up a Workarea”
- “Organizing a Local Working Directory”

Summary of Software/Configuration Requirements

When you complete the procedures in the remaining sections of this chapter, your workstation will be configured with the software required to build, view, and archive (internal to SGI) online books. The following items are required:

IRIX 6.2 or 6.5 The base operating system. The IRIS InSight Professional Publisher tools will not run under IRIX 5.2 or earlier, or under IRIX 6.1. Although there should be no problem installing the IRIX 6.2 subsystems on IRIX 6.3 or 6.4 systems, no testing has been done by SGI to ensure the tools will work.

FrameMaker You must create your documents using FrameMaker 3.1.X or later in order to use the tools or generate SGML with an SGML editor.

bookbuild tools

The translation and figure-processing tools that convert FrameMaker files and associated images to an InSight-viewable format. (This conversion process is called a *bookbuild*.) These tools are part of the IRIS InSight Professional Publisher distribution, in the *insight_dev.sw.tools* subsystem.

- smake* utility A parallel version of *make* that includes additional features, a program that uses a specified rule set to invoke the appropriate translation and figure-processing tools under various bookbuilding conditions. *smake* allows multiple tool invocations at once, which lets your book build faster. *smake* is automatically invoked when you use any of the *make* commands documented in this guide, so you can pretend you're just using *make*. The *make* and *smake* utilities are probably already installed on your workstation; they're part of the software subsystem *dev.sw.make*. To use these utilities, you also need the *irix_dev.sw.headers* subsystem (for IRIX 6.5) or *oe.hdr.lib* subsystem (for IRIX 6.2).
- IRIS InSight software The IRIS InSight viewer, support files for IRIS InSight, and help files for InSight and SGI Help. This software should already be installed on your workstation; it's part of the standard IRIX distribution. The IRIS InSight viewer occupies about 2 MB of disk space without books.
- other tools You need the following additional software subsystems installed in order to make full use of the bookbuilding tools: *oe.sw.imagetools* (basic figure processing tools), and the PostScript-processing and printing tools in *impr_rip.sw.impr* and *x_oe.sw.Xfonts*.
- revision control software SGI publications groups check files into and out of a document archive using the tools in the *oe.sw.rcs* subsystem. ptools are based on RCS (the Revision Control System). If your company uses such a system, you'll need the appropriate revision-control software. This book assumes that you have some sort of system for checking files into and out of an archive; if you don't use such a system, just ignore any references to checking in or checking out files.

Setting Up Tools and Environment

To make a book viewable online, your workstation must have access to the tools that process FrameMaker files into files that can be viewed with IRIS InSight or SGI Help. In addition, your workstation must be running IRIS InSight software so that you can check the online book for problems during development.

Installing Subsystems

The *make* and *impr_rip* subsystems must be installed on your local system. The *insight_dev* subsystem can be installed on your local system or symbolically linked to your workarea. The *prod_toolroot* subsystem, which contains productivity tools for use by SGI production editors and is an SGI-internal product, can be installed on your local system or symbolically linked to your workarea. All of the tools described in this section should be installed in the standard default location of /.

Subsystems for IRIX 6.2 Workstations

Table 2-1 lists the subsystems to install on a local system, or on a local system and a system symbolically linked to your workarea, if the system(s) are running IRIX 6.2, along with the distribution locations for the subsystems.

Table 2-1 Subsystems to Install on IRIX 6.2 Systems

IRIX 6.2	SGI Distribution Location	CD Name and Distribution Location
dev.sw.make	dist.engr:/released/6.2/products/ irix-6.2-development-libraries/ dev	irix-6.2-development-libraries/dev
eoe.hdr.lib	dist.engr:/released/6.2/products/ irix-6.2-part-1-of-2/eoe	irix-6.2-part-1-of-2/eoe
eoe.sw.imagetools	dist.engr:/released/6.2/products/ irix-6.2-part-1-of-2/eoe	irix-6.2-part-1-of-2/eoe
eoe.sw.rcs ^a	dist.engr:/released/6.2/products/ irix-6.2-part-1-of-2/eoe	irix-6.2-part-1-of-2/eoe
impr_rip.sw.impr	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/ impr_rip	irix-6.2-applications-11-98/ impr_rip
x_eoe.sw.Xfonts	dist.engr:/released/6.2/products/ irix-6.2-part-1-of-2/x_eoe	irix-6.2-part-1-of-2/x_eoe
insight_base.sw.eoe	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/ insight_base	irix-6.2-applications-11-98/ insight_base

Table 2-1 (continued) Subsystems to Install on IRIX 6.2 Systems

IRIX 6.2	SGI Distribution Location	CD Name and Distribution Location
insight.sw.xhelp_dev	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/ insight	irix-6.2-applications-11-98/ insight
insight.sw.client	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/insight	irix-6.2-applications-11-98/insight
insight.sw.data	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/insight	irix-6.2-applications-11-98/insight
insight.sw.sgihelp	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/insight	irix-6.2-applications-11-98/insight
insight_dev.sw.tools	dist.engr:/released/6.2/products/ irix-6.2-applications-11-98/../../dev/ insight_dev	irix-6.2-applications-11-98/../../dev/ insight_dev

a. For writers using Revision Control System

Subsystems for IRIX 6.5 Workstations

Table 2-2 lists the subsystems to install on a local system, or on a local system and a system symbolically linked to your workarea, if the system(s) are running IRIX 6.5, along with the distribution locations for the subsystems.

Table 2-2 Subsystems to Install on IRIX 6.5 Systems

IRIX 6.5 Subsystem	SGI Distribution Location	CD Name and Distribution Location
dev.sw.make	dist.engr:/released/6.5/products/ irix-6.5-development-libraries/ dev	irix-6.5-development-libraries/dev
irix_dev.sw.headers	dist.engr:/released/6.5/products/ irix-6.5-development-libraries/ irix_dev	irix-6.5-development-libraries/ irix_dev
eo.e.sw.imagetools	ist.engr:/released/6.5/products/ irix-6.5-foundation-1/eoe	irix-6.5-foundation-1/eoe
eo.e.sw.rcs ^a	dist.engr:/released/6.5/products/ irix-6.5-foundation-1/eoe	irix-6.5-foundation-1/eoe

Table 2-2 (continued) Subsystems to Install on IRIX 6.5 Systems

RIIX 6.5 Subsystem	SGI Distribution Location	CD Name and Distribution Location
x_eoe.sw.Xfonts	dist.engr:/released/6.5/products/ irix-6.5-foundation-1/x_eoe	irix-6.5-foundation-1/x_eoe
impr_rip.sw.impr	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/ impr_rip	irix-6.5-applications-02-99/ impr_rip
insight_base.sw.eoe	dist.engr:/released/6.5/products/ irix-6.5-foundation-1/insight_base	irix-6.5-foundation-1/insight_base
insight.sw.client	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/insight	irix-6.5-applications-02-99/insight
insight.sw.data	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/insight	irix-6.5-applications-02-99/insight
insight.sw.sgihelp	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/insight	irix-6.5-applications-02-99/insight
insight.sw.xhelp_dev	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/insight	irix-6.5-applications-02-99/insight
insight_dev.sw.tools	dist.engr:/released/6.5/products/ irix-6.5-applications-02-99/./dev/	irix-6.5-applications-02-99/./dev/

a. For writers using Revision Control System

To check if any of the subsystems are already installed on the local system, enter

```
% versions dev irix_dev eoe impr_rip x_eoe insight insight_base
insight_dev
```

Compare the output to the subsystems listed in Table 2-1 or Table 2-2. If any of the subsystems are not listed, they need to be installed. Follow the instructions in “Installing Subsystems Using Inst” or “Installing Subsystems Using Software Manager” on page 14 to install the missing subsystems.

Installing Subsystems Using Inst

To install any subsystem using *inst*, the command line installation tool, perform the following steps:

Note: If you need help using *inst*, see your system administrator or the book, *IRIX Admin: Software Installation and Licensing*.

1. Become superuser (root).

```
% su
```

2. To install the subsystem, enter

```
# inst -f /<path>/<to>/<installable>/<image>
Inst> keep *
Inst> list
Inst> i <subsystem_name>
Inst> list i
```

3. Confirm that the subsystem(s) you want to install are marked for installation. Then enter

```
Inst> go
```

4. When the installation is complete, quit *Inst*.

```
Inst> q
```

Installing Subsystems Using Software Manager

To install any subsystem using the Software Manager, the standard SGI graphical installation utility, perform the following steps:

Note: If you need help running *swmgr*, see your system administrator or the *Personal System Administration Guide*.

1. Become superuser (root).

```
% su
```

2. To execute Software Manager, enter

```
% swmgr
```

or

Choose System > Software Manager from the Toolchest

3. Specify the location of the software distribution in the Available Software text field, then click the *Lookup* button.
4. Click the *Customize Installation* button. After a brief processing period, a list of software appears.
5. Select Software > *Short Product Names*, if it isn't already selected. (If it is, leave it selected.)
6. Click the folded arrow icon to display a list of subsystems. The folded arrow icon appears between the *Install* checkbox and the product name.
7. Select the subsystems you need to install by clicking the *Install* checkbox. For example, if you want to install the IRIS InSight software subsystems, check:
 - *insight.sw.client*
 - *insight.sw.data*
 - *insight.sw.sgihelp*
8. Click the *Start* button to install your selections.
9. Click the *OK* button when the successful installation message comes up.
10. Exit Software Manager by selecting File > Quit.

Setting Up a Workarea

A workarea contains the directories of all the books you are writing or have in production. Figure shows a typical SGI workarea.

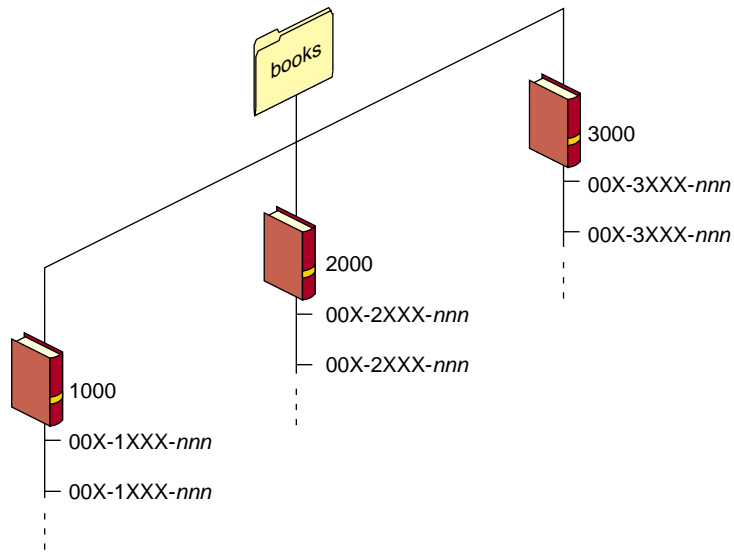


Figure 2-1 A Typical SGI Workarea

Organizing a Local Working Directory

A working directory contains working copies of the files that compose a book, as well as the subdirectories and support files that are needed to produce hard copy and online versions of the book (see Table 2-3). Such a directory is often called a “local working directory” in environments which use a document archiving system; in such an environment, you copy the archived document into a working directory on your local system and make any changes to that local copy.

You may put extraneous files in a working directory and create any extra subdirectories that you wish; however, all files that are actually part of the book must be named according to prescribed naming conventions. These conventions ensure successful book builds and smooth handoffs to other people who need to use your files.

Note: Refer to the information in this section throughout the development process to be sure that your working directory contains the required set of files and subdirectories in the correct location.

Standard Files in a Local Working Directory

Figure 2-2 illustrates the source files that compose the hard and soft copy versions of a technical manual. These files should be archived periodically during the book development cycle. The asterisk (*) indicates files and directories that are unnecessary for books that ship only in hard-copy format.

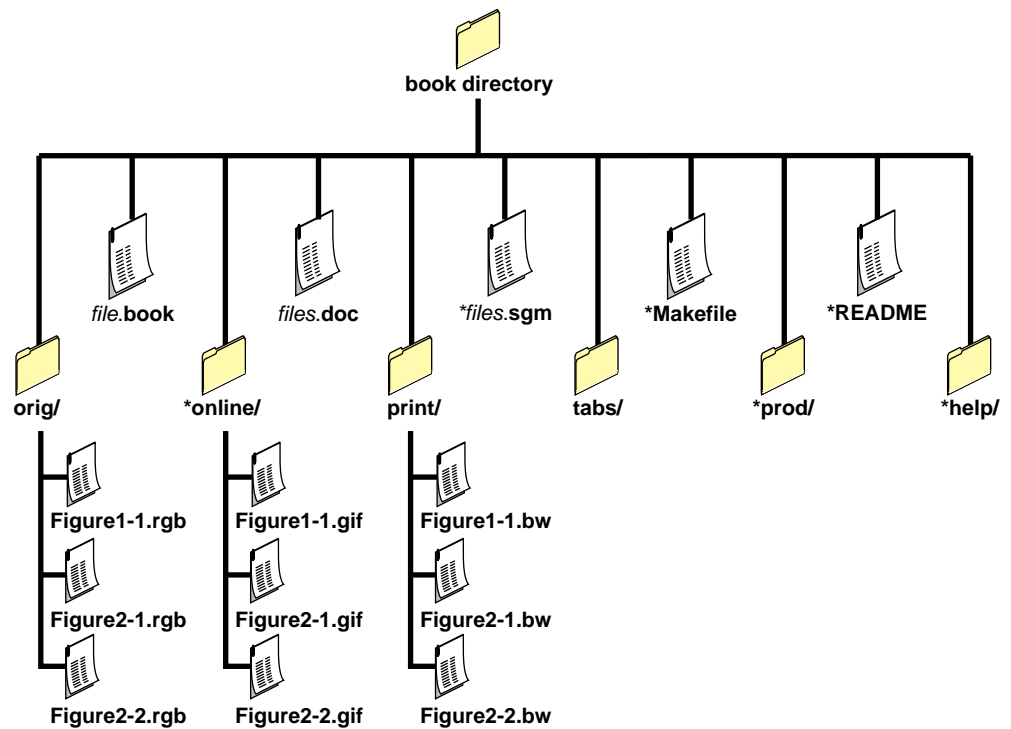


Figure 2-2 Files and Directories in a Working Directory

Table 2-3 describes the files and directories shown in Figure 2-2. The asterisk (*) indicates files and directories used only for online books.

Table 2-3 Files and Directories in a Book Directory

File	Description
<i>print</i> directory	Contains the images used in figures for the printed version of the book. To generate and populate the <i>print</i> directory, use the command <i>make _print</i> . All figures in your FrameMaker files should be imported by reference from this directory.
<i>orig</i> directory	Contains the original images for all figures in the book. Files are either complete figures or figure components that must be processed before storing in the <i>print</i> or <i>online</i> directories.
* <i>online</i> directory	Contains the figures for the online version of the book, generated during the build process.
* <i>help</i> directory	Contains the helpmap file(s) for linking SGI Help to the book. See Chapter 8, "Creating Online Help," for details.
<i>tabs</i> directory	Contains <i>chapter tabs</i> for the book. <i>Part tabs</i> , if any, must be in the same directory as the document files.
FrameMaker document files (with <i>.doc</i> or <i>.fm</i> suffix)	Includes all FrameMaker files for the book: front matter, introduction, chapters, appendices, glossary, and all generated files (TOC, LOF, LOT, and IX). Note that generated files from FrameMaker 5.X have a <i>.fm</i> suffix.
<i>filename.book</i>	The FrameMaker file that contains information about all component files for a book.
*SGML files (with <i>.sgm</i> suffix)	The SGML translations of the <i>.doc</i> or <i>.fm</i> files; <i>.sgm</i> files are the components of the online version of your book. Building a book (with the <i>make</i> command) generates these files.
* <i>Makefile</i> file	Specifies information about how to build your book. The <i>Makefile</i> is used to generate the SGML files for the online version of the book, as well as to generate the various forms of the figure files.
* <i>README</i> file	Contains the writer's name, the date, and the full title, part number, and short title of the book. In addition, gives any other information about the book that might be helpful to future writers, production specialists, or build engineers.
* <i>prod</i> directory	Contains information that might be used by the writer or production specialist for purposes such as storing print vendor specifications.

Observing Naming Conventions

Observe these conventions for naming FrameMaker files:

- All composed FrameMaker document files (front matter, introduction, chapters, appendices, and glossary) must end with the suffix *.doc* or *.fm*. Generated files (TOC, LOF, LOT, LOE, and IX) may end with the suffix *.fm*. See Table 2-4 for guidelines for when to use which suffixes.
- The front matter file should include the string “front” in its name (*PSAGfront.doc*, for example)
- The file containing the introduction to the book—called “About This Guide” in SGI documents—should include the strings “about” or “intro” in its name (*00.about.doc*, for example)
- Chapter filenames should include the chapter number in their names. (*01.getstart.doc*, for example, might be a good name for the first chapter of a book.) Starting the filename with the chapter number means that the filenames appear in the correct order in file-open dialog boxes, when listed by *ls*, and so forth. Including a mnemonic word in the file’s title can be very helpful if you subsequently add or remove chapters in the middle of a book. Thus, filenames like *09.debug.doc* are preferable to names like *ch9.doc*.
- Appendix filenames should include the appendix letter in their names (*C.variables.doc*, for example, for the third appendix in a book). Again it’s useful to start with the sequence letter and to include a key word, rather than using filenames like *AppA.doc*.
- To name your book file, first choose a title for your book, then abbreviate it to form a *short title*. See “Choosing a Book’s Title” and “Choosing a Book’s Short Title” for help with this choice. Name the book file *short_title.book*, where *short_title* is the short title you’ve chosen for the book.
- Your FrameMaker-generated TOC, LOF, LOT, and Index files must end with the suffix *.doc* or *.fm*. For example, if your book file is called *MyBook.book*, the generated files are *MyBookTOC.fm*, *MyBookLOF.fm*, *MyBookLOT.fm*, and *MyBookIX.fm*. FrameMaker chooses these filenames by default when you generate the files within a book.

Table 2-4 Guidelines for Determining Makefile and File Suffixes

Rules to Use	DTD ^a	Makefile	FrameMaker File Suffix	Languages and Encodings Supported
commonbookrules commonbookdef	SGIDOC	Makefile_sgml.doc	.doc	English German French Japanese (EUC)
commondocrules commondocdef	SGIDOCBK or SGIDOC	Makefile_sgml.docbk	.doc or .fm	Same as above plus Japanese (EUC and SJIS) Korean Traditional Chinese (BIG5 and UCNS) Simplified Chinese (GBK and UGB)

a. The functionality of the two DTDs is essentially the same; the online appearance is different, especially for character tags. Before converting all your bookbuilding to the new SGIDOCBK DTD, you'll want to perform a sample bookbuild and view it in IRIS InSight to ensure its online appearance more closely matches your company's style guide than the appearance created by SGIDOC.

Choosing a Book's Title

For consistency, follow standard conventions in naming a book. SGI has guidelines on choosing a title for any kind of document, including many unusual cases. Fortunately, most titles follow a fairly simple structure:

product_name audience_type document_type

The *audience_type* indicates who the book is aimed at, or the kind of task that the book describes. Choices include (among other possibilities):

- Owner's
- User's
- Language
- Programmer's
- Developer's
- System Administrator's
- Installation

The *document_type* indicates what kind of book it is:

- Overview
- Tutorial
- Guide
- Reference Manual
- Instructions

By mixing and matching from the two lists, you should be able to easily choose a title. Here are some examples of SGI book titles:

- *Octane Workstation Owner's Guide*
- *MineSet 3.0 Enterprise Edition User's Guide*
- *C Language Reference Manual*
- *ImageVision Library Programmer's Guide*
- *IRIS Volume Manager System Administrator's Tutorial*
- *Motif Porting Guide*
- *WebMagic User's Guide*

Choosing a Book's Short Title

A short title is an abbreviated way to refer to a book. A book's short title should be a condensed version of its full title, no more than twelve characters long. Short titles are used in a variety of ways: as the titles of icons representing iconified IRIS InSight books; to refer to other books when making cross-book links; and to identify books which contain an item a user has searched for in the main IRIS InSight library window.

Short titles follow conventions similar to those used in the full titles of books; for instance, the short title of a user's guide should end with "_UG", while the short title for a programmer's guide should end with "_PG." Also, the short title of a book is used to identify its subdirectory in the IRIS InSight book directories, so you can't use spaces or other characters that are illegal in a filename (such as * or /).

Be sure to list your book's short title on the TITLE line of your *Makefile*, and in your *README* file. See "Editing the Makefile" in Chapter 6 for details.

Note: Do not use the short title of your book as the basis for the name of any file other than the book file. For example, *short_title.doc* (where *short_title* is the short title of your book) is not a valid name for a chapter file.

Copying Template Files to Your Working Directory

IPTemplates

Use the template files provided with IRIS InSight Professional Publisher to create new FrameMaker files for your book. These template files are located in the directory `/usr/share/Insight/templates/frame/IPTemplate`. This directory also contains a subdirectory called *samples* which contains a sample book that uses the templates. If you're unfamiliar with this template, start by launching FrameMaker and open the IP.book. Each file is a self-documenting instruction set that includes information on all tags and formatting procedures for FrameMaker documents that will be converted to IRIS InSight.

Makefile

In addition to FrameMaker template files, the bookbuilding tools also require a *Makefile*. This file specifies the characteristics necessary to prepare a document for online viewing—particularly information about what files are part of the document to be built.

If you are building your book with the SGIDOCBK DTD, copy the file `/usr/share/Insight/templates/make/Makefile_sgdocbk` into your working directory, rename it to *Makefile*, and then set the permissions so you can both read and write it:

```
% cd working_directory
% cp /usr/share/Insight/templates/make/Makefile_sgdocbk ./Makefile
% chmod 644 Makefile
```

If you are building your book using the older SGIDOC DTD, copy the file `/usr/share/Insight/templates/make/Makefile_sgldoc` into your working directory, rename it to *Makefile*, and then set the permissions so you can both read and write it:

```
% cd working_directory
% cp /usr/share/Insight/templates/make/Makefile_sgldoc ./Makefile
% chmod 644 Makefile
```

Note: The functionality of the two DTDs is essentially the same; the online appearance is different, though, especially for character tags. Before converting all your bookbuilding to the new SGIDOCBK DTD, you'll want to perform a sample bookbuild and view it in IRIS InSight to ensure its online appearance more closely matches your company's style guide than the appearance created by SGIDOC.

Once you've set up your *Makefile*, you have to edit it to make it work with your particular book. Directions for editing the *Makefile* are given in Chapter 6, "Building Online Books."

Using the IP Templates to Author Documents

This chapter provides an overview of the structure and usage of the paragraph and character tags used in the IRIS InSight Professional Publisher Templates. It also explains how to use SGI's implementation of certain FrameMaker features, and how to generate and format a table of contents and lists of examples, figures, and tables. The sections in this chapter include

- "IP Templates - A Quick Tutorial" on page 26
- "Applying the Templates to Existing Files" on page 29
- "Paragraph Tag Naming Scheme" on page 31
- "Chapter and Appendix Numbers and Titles" on page 31
- "Headings" on page 31
- "Lists" on page 32
- "Text Paragraph Tags" on page 37
- "Code and Example Paragraph Tags" on page 38
- "Messages and Explanations" on page 42
- "Checkoff Paragraph Tags" on page 43
- "Equation Paragraph Tags" on page 43
- "Figure Tags" on page 46
- "Tables" on page 47
- "Hint, Tip, Note, Caution, Warning, and Shortcut Paragraph Tags" on page 50
- "Flag Paragraph Tags" on page 57
- "Footnote Paragraph Tag" on page 60
- "Page Break Paragraph Tag" on page 60
- "Character Tags" on page 60
- "Cross References" on page 62

- “Markers” on page 63
- “Conditional Text” on page 64
- “InSight Inline Object Tags” on page 64
- “Help Paragraph Tags” on page 64
- “Creating a Book File” on page 65

IPTemplates - A Quick Tutorial

The IPTemplates are a set of FrameMaker files pre-formatted with text margins, headers, page-numbered footers, paragraph and character tags, table formats, conditional text settings, variables, and markers which can be translated into SGML by the bookbuild tools. The IPTemplates are made up of the following files:

- “IPBook” on page 26
- “IPFront.doc” on page 27
- “IPTOC.doc, LOE.doc, LOF.doc, LOT.doc, and IX.doc” on page 27
- “IPIntro.doc” on page 28
- “IPPartTab.doc and IPTab.doc” on page 28
- “IPChap.doc and IPNoChap.doc” on page 28
- “IPAppendix.doc” on page 28
- “IPGlossary.doc” on page 29

IPBook

The book file is a list of filenames, each name referencing a file you wish to include in your book. The names are listed in the order in which the files should appear in your book, and attached to each name (though not appearing in the main book file window) is a collection of information about how the file will be paginated, and how figures, tables, and heading paragraphs are numbered. The book file contains both files you create through standard text entry (chapters, appendixes, and so on), and special “generated” files (Table of Contents, List of Figures, List of Tables, Index) that you create and update using utilities in the book file.

When you access the File menu through the book file, the selections are slightly different than for a normal file. This is because the book file is not a true text file that you can edit in the normal manner, but rather an organizational and formatting tool for your book as a whole. When you want to perform operations that affect how the book file orders, paginates, or generates and updates the files it contains, use the book file. If you wish to alter the content of a text file, or change the output of a generated file, open that particular file.

For information on creating and populating a bookfile, see “Creating a Book File” on page 65.

IPFront.doc

The *IPFront.doc* contains the title page, credits page, a “what’s new” section and a record of revision section. Customize the boilerplate information, such as book title, document number, credits, copyright, limited and restricted rights legend, regulatory information, and trademarks, taking care to keep the paragraph tagging the same as the boilerplate. If your book is a revision of an existing book, complete the “What’s New in this Guide” and “Record of Revision” sections. If it is a new book, remove these sections from the file.

The *IPFront.doc* should be named *front.doc* for your book and be the first file listed in the bookfile.

IPTOC.doc, LOE.doc, LOF.doc, LOT.doc, and IX.doc

Once you have created a bookfile and have at least one file listed in it, you can generate a Table of Contents, List of Examples, List of Figures, List of Tables, and an Index from the File > Generate/Update utility. These templates are applied *after* the generated files are created and provide the necessary formatting and structure to the generated information. See “Applying the Templates to Existing Files” on page 29 for instructions on how to apply the IPTemplates to your generated files.

To create generated files, see “Adding Generated Files to Your Book File” on page 69.

IPIntro.doc

The *IPIntro.doc* template is used for a preface and/or the “About This Guide” portion of your manual. For books with no part tabs or chapter tabs, the *IPIntro.doc* is positioned in the bookfile directly before the first chapter. When a part tab and/or chapter tab is used, the *IPIntro.doc* immediately proceeds the tab(s). Since the preface or “About This Guide” is considered part of the front matter of a book, the page numbering continues in roman numeral format.

IPPartTab.doc and IPTab.doc

Tabs are optional divider pages that appear at the beginning of a printed book’s chapters, or, in the case of online and printed books, at the beginning of a book’s parts. Position the tabs in the bookfile accordingly. Do not use tabs before an appendix or a group of appendices. No page numbers appear on tabs; however, the book pagination accounts for their presence.

IPChap.doc and IPNoChap.doc

The *IPChap.doc* is used for all chapters of a multi-chapter book. For a one-chapter book, use the *IPNoChap.doc*, where a chapter title but no chapter number appears at the top. Follow the guidelines in the *SGI Editorial Style Guide* or your company’s own style guide when writing the content of your chapters, and use the paragraph and character tags as described in this book for formatting your content.

IPAppendix.doc

The *IPAppendix.doc* is used for one or more appendices in your book and include an appendix title and identifying letter. Follow the guidelines in the *SGI Editorial Style Guide* or your company’s own style guide when writing the content of your appendices, and use the paragraph and character tags as described in this book for formatting your content.

IPGlossary.doc

If new terminology is introduced in your book and it is not included in the *SGI Glossary of Terms*, the *IPGlossary.doc* can be used to provide the glossary terms and definitions. When a user clicks on a glossary item in the online version of your book, the glossary's definition appears in a restricted viewer. For books with a local glossary, the definition in the local glossary is accessed. If the glossary item doesn't have a local glossary definition, then the definition in the master glossary appears. Also use the *IPGlossary.doc* if you want a complete glossary for your book.

Applying the Templates to Existing Files

To apply any IP template to a document file:

1. Open and iconify the appropriate template file.
2. Open the file to which you are applying the template.
3. Delete all paragraph tags.
 - Open the paragraph catalog by clicking on the ¶ icon.
 - Click *Delete*.
 - When the Delete Formats dialog box appears, place your cursor over the *Delete* button, click it once, and hold down the Enter key.
 - When all paragraph tags are gone, click *Done*.
4. Delete all character tags.
 - Open the character tag catalog by clicking on the f icon.
 - Click *Delete*.
 - When the Delete Formats dialog box appears, place your cursor over the *Delete* button, click it once, and hold down the Enter key.
 - When all character tags are gone, click *Done*.

5. Delete all table formats.
 - Choose Table > Table Designer > Commands > Delete Format.
 - When the Delete Formats dialog box appears, place your cursor over the *Delete* button, click it once, and hold down the Enter key.
 - When all table formats are gone, click *Done*.
6. Delete all master pages except for Right, Left, and First.
 - Choose View > Master Pages.
 - Choose Special > Delete Page <master page name>.
7. Delete reference pages.
 - Choose View > Reference Pages.
 - Choose Special > Delete Page <reference page name>.
8. Delete *unused* cross-reference formats.
 - Choose Special > Cross-Reference.
 - Click the *Edit Format...* button.
 - When the Edit Cross-Reference Format dialog box appears, place your cursor over the *Delete* button, click it once, and hold down the Enter key.
 - When all cross-reference formats are gone, click *Done*.
 - Click *Cancel* for all warning messages that appear asking if you would like to change all cross-references of format *x* to editable text.
 - Click *Done* when you return to the main cross-reference window.
9. Apply the IPtemplate formats.
 - Select File > Import > Formats.
 - When the Import Formats dialog box appears, select the iconified IPtemplate from the “Import From Document:” popup menu.
 - Select all the formats in the “Import and Update” list.
 - Click *Import*.
10. Choose File > Save.

Paragraph Tag Naming Scheme

The length of the paragraph tag list is more a result of the number of variations within basic tag groups than the number of groups themselves. These variations are necessary to properly indent paragraphs, according to this indentation scheme:

- Paragraphs at the same level should be indented the same amount.
- The left margin of the second paragraph in a list item should be indented such that it lines up with the *left text margin* of the previous paragraph (see “Lists” on page 32 for examples).

Indentation levels for tags within a group use a standard nomenclature: *Tagname* for the tag that puts the paragraph at the margin; *TagnameInd* or *TagnameInd1* for the first 1/4" indent, and *TagnameInd2*, *3*, *4*, and so on for each succeeding 1/4" indent.

Note: Headings and introductory matter (*AppNum*, *ChapNum*, *ChapTitle*, *GlossaryTitle*) are exceptions to this scheme.

Chapter and Appendix Numbers and Titles

The introduction, chapters, appendixes, and index have one, and only one, *ChapTitle* tag. The glossary uses a similar tag, *GlossaryTitle*. In an introduction, appendix, glossary, or index, the title tag is the first tag in the file. In chapters and appendixes, *ChapTitle* is preceded by *ChapNum* and *AppNum*, respectively. In the introduction, chapters, and appendixes, the *ChapTitle* tag should always be followed by at least one *Text* tag, the contents of which introduce the material.

Headings

The template supports three levels of headings: *Heading1*, *Heading2*, and *Heading3*. The standard rule for nesting sections is that there should never be only one of any section level nested within the next highest section. This means that *Heading1*, for example, should be followed by two or more *Heading2*s, or none at all.

Note: It is recommended that you only use three levels of headings. The online translator includes support for *Heading4* and *Heading5*; however, these tags exist to support legacy documents in which these paragraph tags were used in SGI documents.

Second-Level Heading

This is a dummy paragraph.

Third-Level Heading

This is a dummy paragraph.

Third-Level Heading

This is a dummy paragraph.

Second-Level Heading

This is a dummy paragraph.

Lists

There are six types of lists: bulleted, square-bulleted, dashed, ordered, hanging, and tabular. Lists may appear at the margin or within other lists (given the specific restrictions described below). However, nesting beyond one level (that is, nesting two or more levels deep) is not supported in the templates. Lists may not appear within Notes, Cautions, Warnings, Hints, Tips, or Shortcuts.

These further restrictions apply to lists:

- A list must contain at least two items.
- A bulleted list should not appear within a bulleted list. Instead of the nested bullets, use a dashed list for the indented material.
- A dashed list appears only within a bulleted list.
- A hanging list may not appear within another hanging list.
- The square-bulleted list for is used only for substeps within an ordered list.
- No sublists may appear inside tabular lists.

Bulleted Lists

There are three tags for bulleted lists: *Bullet*, *BulletInd*, and *BulletInd4*.

This is the *Bullet* tag:

- Bullet
- Another Bullet

BulletInd appears here within a numbered list:

1. First list item
 - BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd BulletInd
 - Another BulletInd
2. Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item Second List Item

BulletInd4 is used here within a hanging list:

- | | |
|----------|---|
| HangBody | HangItem |
| | <ul style="list-style-type: none"> • BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 BulletInd4 • Another BulletInd4 |
| HangBody | HangItem |

Square Bulleted Lists

Use a square bulleted list for substeps within a numbered list.

1. First Step
 - *BulletSquareInd* paragraph tag indicates a substep substep substep substep substep substep substep substep substep substep
 - next substep
 - last substep
2. Next Step

This example uses the *ListInd4* paragraph tags:

```

HangItem      HangBody
1.  ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st
    ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st
    ListInd4.1st ListInd4.1st
2.  ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4
    ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4
    ListInd4 ListInd4 ListInd4 ListInd4
HangItem      HangBody
  
```

Hanging Lists

There are two hanging list tags: *HangingList* and *HangingListInd*. If the options or commands you wish to document are longer than the space provided, use a tab and then a forced return (Shift+Enter, Control+J, Alt+Enter, or Alt+M) to begin a new line. Do not alter the paragraph tag.

Here is a sample hanging list:

```

Item          body of the hanging list item body of the hanging list item body of the
              hanging list item body of the hanging list item body of the hanging list
              item body of the hanging list item
another item  body of the second hanging list item
  
```

Here is another sample hanging list. This one uses *HangingListInd* and a forced return after the tab for each item because at least one of the items exceeds the length of the tab delimiter.

- Bullet
 - thisisanitemofarbitrarylength
 - used to demonstrate what hanging lists should look like when the
 item to be documented hangs over the explanatory text.
 - shorter item
 - another hanginglist body.
- bullet

Tabular Lists

Tabular lists are multi-column tables that do not use the explicit markings (such as headings and titles) that one normally associates with tables. They are either multi-column lists or hanging lists where the item in the first column wraps. They are created using the FrameMaker table utility. Use one of the table formats called *TabularList*. For the text in a tabular list, use the paragraph tag *TabularListText*. Anchor the table at the end of the preceding paragraph.

This is a table created with the *TabularList* format:

item one	item four	item seven
item two	item five	item eight
item three	item six	item nine

This is bulleted list with an indented hanging list created with the *TabularList* format:

- Bullet
 - item one Long explanation of item one. Long explanation of item one. Long explanation of item one. Long explanation of item one. Long explanation of item one. Long explanation of item one.
 - item two is a long one that wraps and wraps and wraps Long explanation of item two. Long explanation of item two. Long explanation of item two. Long explanation of item two. Long explanation of item two. Long explanation of item two. Long explanation of item two.
- Bullet

This is a tabular list inside a hanging list:

HangItem	HangBody		
	r1c1	r1c2	r1c3
	r2c1	r2c2	r2c3
	r3c1	r3c2	r3c3

Text Paragraph Tags

Text is the standard unadorned paragraph tag. Text always appears at least once following a *ChapTitle* or *HeadingX*. It can be used within any section, interspersed with Code, Example, Hint, Tip, Note, Caution, Warning, or any type of list. It can also appear as a second and following paragraph in any type of list. There are five levels of indentation for regular text: *Text*, *TextInd*, *TextInd2*, *TextInd4*, and *TextInd5*.

This sentence is *Text*, and the tag in the bulleted item is *TextInd*:

- Bullet

TextInd TextInd TextInd TextInd TextInd TextInd TextInd TextInd TextInd TextInd
TextInd TextInd TextInd TextInd TextInd TextInd

- Bullet

The example below uses *TextInd2*:

- Bullet

- Dash

TextInd2 TextInd2 TextInd2 TextInd2 TextInd2 TextInd2 TextInd2 TextInd2
TextInd2 TextInd2 TextInd2 TextInd2 TextInd2 TextInd2

- Dash

- Bullet

The following example uses *TextInd4* and *TextInd5*:

HangItem HangBody

TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4
TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4
TextInd4 TextInd4 TextInd4 TextInd4

HangItem HangBody

- BulletInd4

TextInd5 TextInd5 TextInd5 TextInd5 TextInd5 TextInd5 TextInd5
TextInd5 TextInd5 TextInd5 TextInd5 TextInd5 TextInd5 TextInd5
TextInd5

- BulletInd4

Code and Example Paragraph Tags

The *Code* and *Example* tags both produce Courier text. The only difference between the two is the amount of spacing between paragraphs: *Code* has no line spacing, *Example* has 5 points of line spacing. *Code* is used mainly for blocks of programming code or output, where it's important to fit as much on the page (or screen) as possible. *Example* is used primarily for single code lines (*Code* and *Example* are basically interchangeable in this context) or for sequences of command lines. *Code* and *Example* can appear at the margin, or within lists, but not within Hints, Tips, Notes, Cautions, Warnings, or Shortcuts.

Code and Example tags do not contain any tabs. Do not add any tabs. If you do, you run two risks: Using File > Import > Formats will overwrite them, and your code will not be formatted correctly if your book is built for online use. If you need to import code or example text that contains tabs, there are scripts available to convert the tabs to spaces *before* you import the text in FrameMaker.

Note: When you import code or examples, be sure to turn off *Smart Spaces* in the Format > Document > Text Options dialog box before you begin the process. Otherwise, FrameMaker will eliminate all multi-space blocks in your imported text, and replace them with single spaces.

Caution: Once you begin a block with an *Example*, *Code1st*, *CodeExt1st*, or *CodeMargin1st* paragraph, you must tag subsequent paragraphs in the block with the appropriate variation of that tag for second and following paragraphs. If you mix and match Code and Example tags, the online tools will mis-format your block.

Example Paragraph Tags

When you need to embed a line or two of code in regular text, use an Example tag. There are five variations: *Example*, *ExampleInd*, *ExampleInd2*, *ExampleInd4*, and *ExampleInd5*.

When you need to embed a line or two of code in regular text, use the *Example* paragraph tag:

`Example`

The following example uses *ExampleInd*:

- Bullet
ExampleInd
- Bullet

The following example uses *ExampleInd2*:

- Bullet
 - Dash
ExampleInd2
 - Dash
- Bullet

The following example uses *ExampleInd4* and *ExampleInd5*:

- HangItem HangBody
ExampleInd4
- HangItem HangBody
 - BulletInd4
ExampleInd5
 - BulletInd4

Code Paragraph Tags

For sample programs—or simply for longer code examples—use a Code paragraph tag. To give your code block the correct vertical spacing above it, use the *1st* paragraph tag for the first line of code. There are ten basic Code tags: *Code1st*, *Code*, *CodeInd1st*, *CodeInd*, *CodeInd2.1st*, *CodeInd2*, *CodeInd4.1st*, *CodeInd4*, *CodeInd5.1st*, and *CodeInd5*. In addition, there are four Code tags for displaying longer lines of code that don't fit within the margins: *CodeExt1st*, *CodeExt*, *CodeMargin1st*, and *CodeMargin*.

Note: If you find it necessary to break a line of code, use the proper syntax for the programming language you are documenting.

The following example uses *Code1st* and *Code*:

```
Code1st  
Code
```

The following example uses *CodeInd*:

- Bullet
CodeInd1st
CodeInd

- Bullet

The following example uses *CodeInd2*:

- Bullet
 - Dash
CodeInd2.1st
CodeInd2

- Dash

- Bullet

The following example uses *CodeInd4* and *CodeInd5*:

```
HangItem    HangBody  
             CodeInd4.1st  
             CodeInd4
```

```
HangItem    HangBody  
             • BulletInd4  
             CodeInd5.1st  
             CodeInd5  
             • BulletInd4
```

If you have long lines of code that require more characters per line, you can use *CodeExt1st* and *CodeExt* tags, which reduce the letter spacing and produce an 80-column line within the text margin:

```
long line of code that wouldn't fit in the text column and uses CodeExt1st tag  
another long line that uses the CodeExt tag to keep it from breaking  
12345678902234567890323456789042345678905234567890623456789072345678908234567890
```

You may also use the *CodeMargin1st* and *CodeMargin* paragraph tags to display long lines of code:

For **real** long lines of code, you might need to use the *CodeMargin1st* paragraph tag. Its companion, for second and following lines, is the *CodeMargin* paragraph tag

Numbered Code and Example Blocks

For code and example blocks that you want to be able to cross-reference and display in the List of Examples at the beginning of your book, precede your block with either the *CodeTitle* or *ExampleTitle* paragraph tag, depending on the paragraphs that make up your block.

Note: Both tags produce the same automatic string preceding the title, and use the same numbering scheme. The differentiation is necessary for online conversion, because Code and Example are different constructs in the SGML DTD.

Example 3-1 Code Block Beginning With the *CodeTitle* Paragraph Tag

```
blah, blah, blah  
blah, blah, blah  
blah, blah, blah
```

Example 3-2 Example Block With the *ExampleTitle* Paragraph Tag

```
% su  
# df -k  
# Ctrl-D  
%
```

Messages and Explanations

There are five message tags: *Msg*, *Msg1st*, *MsgExpl*, *MsgMargin*, and *MsgMargin1st*. These are the tags for things like error messages, system feedback, and so on. This is the sequence for error message constructs:

- one *Msg1st* (or variants) paragraph tag
- zero or more *Msg* (or variants) paragraph tags
- one or more *MsgExpl* paragraph tags
- zero or more paragraph tags that can appear following the paragraph tag *MsgExpl*.

Msg1st

MsgExpl *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl*
MsgExpl *MsgExp* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl*

Msg1st

Msg

MsgExpl *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl*
MsgExpl *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl*

- **BulletInd4** **BulletInd4** **BulletInd4** **BulletInd4** **BulletInd4** **BulletInd4** **BulletInd4**
BulletInd4 **BulletInd4**

ExampleInd5 *ExampleInd5* *ExampleInd5* *ExampleInd5*

TextInd5 *TextInd5* *TextInd5* *TextInd5* *TextInd5* *TextInd5*

Note: *NoteInd5*

Hint: *HintInd5*

TextInd4 *TextInd4* *TextInd4* *TextInd4* *TextInd4* *TextInd4* *TextInd4*

ExampleInd4 *ExampleInd4* *ExampleInd4* *ExampleInd4*

1. *ListInd4.1st* *ListInd4.1st* *ListInd4.1st* *ListInd4.1st* *ListInd4.1st*

TextInd5 *TextInd5* *TextInd5* *TextInd5* *TextInd5* *TextInd5* *TextInd5*

2. *ListInd4* *ListInd4* *ListInd4* *ListInd4* *ListInd4* *ListInd4*

Note: *NoteInd4*

If you need more space, use *MsgMargin1st* and *MsgMargin*.

023456789012345678902234567890323456789042345678905234567890623456789072345678908234567890923
 023456789012345678902234567890323456789042345678905234567890623456789072345678908234567890923

MsgExpl *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl* *MsgExpl*
MsgExpl *MsgExpl* *MsgExpl* *MsgExpl*

TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4

1. ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st ListInd4.1st
 ListInd4.1st ListInd4.1st

2. ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4 ListInd4
 ListInd4 ListInd4 ListInd4

TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4 TextInd4

Note: NoteInd4

Checkoff Paragraph Tags

There are two tags for checkoff boxes:

- Checkoff1st*
- Checkoff*
- another checkoff box

Equation Paragraph Tags

Use the *Equation* paragraph tag to produce equations with titles and numbering. If you don't want the numbering, use the appropriate version of the *Text* tag necessary to produce the proper level of indentation. For full instructions on how to produce equations, see the FrameMaker documentation.

Note: Equations are not currently supported online, except as snapped figures. If your document will be built for online use and you wish to include equations, snap them and treat them as figures as described in “Creating Equations as Figures” on page 74. Apply the conditional text setting `OnlineOnly` to the anchored frame and `PrintOnly` to the equation and the `Equation` paragraph tag. Prior to printing the book, hide the `OnlineOnly` conditional text.

Equation 1

$$m = \frac{\Delta y}{\Delta x}$$

Equation 1(This is the print version.)

$$m = \frac{\Delta y}{\Delta x}$$

Guidelines for Typing Mathematical Equations

Typing math equations is different from typing text. The FrameMaker math editor is also a math package that evaluates and solves equations. SGI does not currently use this equation solving capability, but you should enter equations with the proper syntax to give them a mathematically correct look and to enable use of the equation solver in the future.

Equation objects include variables, operators, relations, and functions. Each entry that you make from the keyboard or from the `Special > Equations` menu has a mathematical meaning. For example, FrameMaker assumes that a single letter is a variable. When you type one letter after another, FrameMaker assumes the variables are to be multiplied, because in mathematical notation, multiplication is implicit when variables are combined with no spaces between them. If what you really intend is a single variable whose name contains two or more characters, you must select *Start String* from the `Equation` menu and type the variable name between the quotes. The quotes disappear when you enter the next object in the equation. Use the space bar to select equation objects for editing.

When you use a variable from an equation in a sentence, some engineers prefer that variable to be set off from the surrounding text with a numeric space before and after it. This prevents the variable-width spacing adjustment from being applied to the spaces surrounding the variable, thus making it stand out from the surrounding text.

Mathematical Symbol Usage Guidelines for All Documents

Even if you don't use equations, keep in mind that the typography of mathematical symbols is different from that of letters and text symbols. For example, a minus sign has a slightly different appearance and vertical alignment from a dash symbol. A multiplication symbol is different from the letter x.

Use the minus symbol and apply the Symbol character tag rather than use a dash to indicate a negative value or a subtraction. Similarly, enter the multiply symbol (Ctrl+Q 4 and apply the Symbol character tag) rather than the letter x when you intend to represent multiplication. The most common case of this is when quoting dimensions, as in 3×5 . Do not use the letter x to represent "3 by 5."

Table 3-1 contains a list of the most common mathematical symbols and how to create them so they appear correctly in your FrameMaker document as well as your online book. See FrameMaker Help's keyboard map for additional symbols.

Table 3-1 Creating Mathematical Symbols

Symbol	Key Combination in FrameMaker	Character Tag to Apply
±	Ctrl+Q 1	Symbol
×	Ctrl+Q 4	Symbol
÷	Ctrl+Q 8	Symbol
–	Dash	Symbol
≥	Ctrl+Q 3	Symbol
≤	Ctrl+Q Shift+3	Symbol
∞	Ctrl+Q Shift+5	Symbol

Figure Tags

There are two tags used for figures that appear inline in the hard copy: *Fig* and *FigTitle*. The *Fig* tag should not have any text in it. Its sole purpose is to serve as a placeholder for the anchored frame that holds the figure. *FigTitle* follows immediately after *Fig*, and holds the title of the figure.

Use this procedure to set up figures:

1. Press Enter to create a new paragraph.
2. Apply the *Fig* paragraph tag to the paragraph.
3. Press Enter again to create a paragraph that automatically has the *FigTitle* paragraph tag applied to it. Type in the figure title.
4. Place the cursor in the *Fig* paragraph and Choose Special > Anchored Frame.
5. In the Anchored Frame dialog box, choose these settings:
 - Anchoring Position: Below Current Line
 - Alignment: Right
 - Cropped
 - Size: width = 5.375" and height = your choice (the frame can be resized once it's inserted)
6. Click *New Frame*.
7. Import your figure *by reference* from the *print* directory into the anchored frame, using the File > Import > File dialog box.

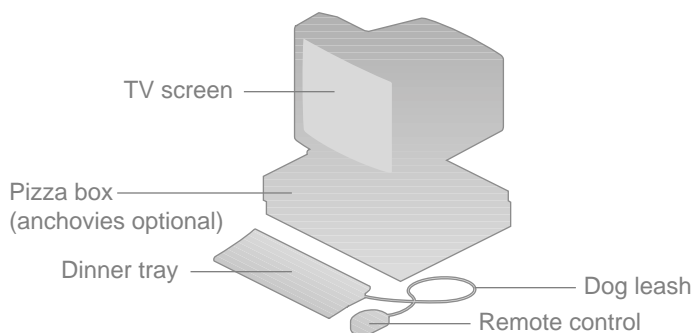


Figure 3-1 Dummy Figure Title Using FigTitle Paragraph Tag

To select a text object in an anchored frame, the anchored frame itself, or a text column on a page, without opening the toolbox and changing to the arrow selection tool, simply hold down Ctrl while clicking the object. Shift selects and deselects additional objects after the first object is selected.

To move graphic objects in incremental (smaller than grid units) steps, even when Graphics > *Snap* is turned on, hold down Ctrl and use the arrow keys to move objects.

For more information about figure formats, see Chapter 5, “Working With Figures.”

Tables

To create a table, follow these steps:

1. Place the cursor at the end of the paragraph that immediately precedes where you want the table to go.
2. Choose Table > Insert Table.
3. Select the proper Table Format and insert the number of body rows you need (you can always add or delete these later). Click *Insert*.

A pre-formatted table with a pre-numbered table title appears just below the text paragraph.

The templates use three paragraph tags for text in the various parts of a table. All these tags are applied automatically when you create your table.

- The table title appears in the table block above the table. It uses the *TableTitle* paragraph tag.

Place your cursor at the end of the table number and press Esc+Tab. Type in your table title.

- Heading Cells use the *TableHead* paragraph tag.

Place your cursor in the first row’s cell and type the first column’s heading. Pressing Tab allows you to navigate from cell to cell.

- Body cells use the *TableText* paragraph tag.

Customizing Tables

Tables can be customized by adding, deleting, or resizing rows and columns as needed.

If you need additional columns or rows:

1. Choose Table > Add Rows or Columns
2. Complete the fields in the dialog box as necessary.
3. Click *Add*.

To delete any unnecessary rows or columns:

1. Highlight the rows or columns.
2. Press Backspace:

The Clear Table dialog box allows you to select *Leave Cells Empty* or *Remove Cells from Table*.

3. Click *Clear*.

If you need to delete an entire table and its table title:

1. Highlight the anchor for the table (the table will appear highlighted as well)
2. Press Backspace.

To resize columns:

1. Highlight the rows or columns.
2. Choose Table > Resize Columns.
3. Complete the fields in the Resize Selected Columns dialog box.
4. Click *Resize*.

Tables should always line up with the left text margin.

Inserting Figures in Tables and Tabular Lists

Figures, such as screen snaps of icons or other small images, can be included in a table cell. To include a figure within a table:

1. Place your cursor in the cell in which you want a figure.
2. Choose Special > Anchored Frame.
3. In the Anchored Frame dialog box, choose these settings:
 - Anchoring Position: At Top of Column
 - Alignment: Left
 - Cropped
 - Size: width = cell width and height = your choice (the frame can be resized once it's inserted)
4. Click *New Frame*.
5. Import your figure *by reference* from the *print* directory into the anchored frame, using the File > Import > File dialog box. See Chapter 5, "Working With Figures," for complete instructions.

Caution: Though you may use any *character* tags you wish within a table, do not use any *paragraph* tags other than those designated for each of the three table parts. In addition, if you include any equations created in Special > Equations in your tables, they will not appear online. Follow the instructions in "Creating Equations as Figures" on page 74 if you want to include equations in your tables.

Table 3-2 uses the *Table-5Col* table format tag found in the Table Format dialog box.

Table 3-2 Real N x N Operations (TableTitle)^a

N	Load 4, 4*N	N	Load 4, 4*N	N
128	.003	128	.003	128
256	.0105	256	.0105	256
512	.039	512 ^b 312	.039	512
128	.003	128	.003	128
256	.0105	256	.0105	256
512	.039	512	.039	512
1024	.150	1024	.150	1024

- a. This is a table footnote in a table head. It uses the *TableFootnote* tag.
- b. This is a table footnote in a table cell. It uses the *TableFootnote* tag.

If you have a multi-page table, the string “(continued)” should appear after “Table X-X.” To force this string to appear, insert the Special > Variable > *TableContinuation* variable before the Tab in your table title. It does not appear in the first page of your table, so you may choose to insert it as a precaution even if your table currently fits on a single page.

Hint, Tip, Note, Caution, Warning, and Shortcut Paragraph Tags

Hints, Tips, Notes, Cautions, Warnings, and Shortcuts are used to provide short, supplemental or cautionary information. They are *not* meant to provide space for long blocks of interpolated information. Therefore, there is no support for second and following paragraphs either in the templates or in the online translator. To highlight these paragraphs, a horizontal rule appears just above and just below the paragraph.

Note: The workaround, if you absolutely must add a line of Code or Example after your Note, is to use two forced returns, like this: <Shift+Enter>
<Shift+Enter>
line of code

Caution Paragraph Tags

Use one of the *Caution* paragraph tags to point out actions that may cause damage to your system. There are five *Caution* paragraph tags: *Caution*, *CautionInd1*, *CautionInd2*, *CautionInd4*, and *CautionInd5*.

This example uses *Caution*:

Caution: Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution Caution

The next example uses *CautionInd1*:

- Bullet
Caution: CautionInd1 CautionInd1 CautionInd1 CautionInd1 CautionInd1 CautionInd1 CautionInd1 CautionInd1 CautionInd1
- Bullet

The following example uses *CautionInd2*:

- Bullet
 - Dash
Caution: CautionInd2 CautionInd2 CautionInd2 CautionInd2 CautionInd2 CautionInd2 CautionInd2 CautionInd2 CautionInd2
 - Dash
- Bullet

The following example uses *CautionInd4* and *CautionInd5*:

- | | |
|----------|---|
| HangItem | HangBody |
| | Caution: CautionInd4 CautionInd4 CautionInd4 CautionInd4 CautionInd4 CautionInd4 CautionInd4 CautionInd4 |
| HangItem | HangBody |
| | <ul style="list-style-type: none">• BulletInd4
Caution: CautionInd5 CautionInd5 CautionInd5 CautionInd5 CautionInd5 CautionInd5 CautionInd5• BulletInd4 |

Adding the Warning Icon to Warning Paragraph Tags

Each of the three warning paragraph tags has an icon to the left of it. This icon must be added manually, following this procedure:

1. Place the cursor at the very beginning of a *Warning* paragraph.
 Choose Special > Anchored Frame and select the correct settings according to the type of warning paragraph tag you are using. See Table 3-3.

Table 3-3 Anchored Frame Specs for Warning Tags

Paragraph Tag	Anchor Position	Baseline Offset	Near-Side Offset	Height	Width
<i>Warning</i>	Left side	-0.2"	-1.5"	0.5"	.05"
<i>WarningInd1</i>	Left side	-0.2"	-1.75"	0.5"	0.5"
<i>WarningInd2</i>	Left side	-0.2"	-2.0"	0.5"	0.5"

When you complete this and reactivate the page, you see an empty box to the left of the Warning.

2. Choose View > Reference Pages and scroll to the Warning Reference Page. Copy the warning symbol in the square box. The icon is clearly labeled and has an explanation with it. After you copy it, choose View > Body Pages.
3. Select the anchored frame you created and use the Edit > Paste command to paste the icon into the frame. *Do not* move the icon around in the frame. It is positioned correctly and moves with the text.

Note: When the Warning paragraph is the first paragraph on the page, FrameMaker bumps the anchored frame down so it doesn't run above the top of the upper margin. You must add an empty Text tag above the Warning tag, then set the Space Above for the Warning paragraph to 0.0". Only employ this workaround when you are certain that a Warning tag will fall at the top of a page.

Shortcut Paragraph Tags

There are two shortcut tags: *Shortcut* and *ShortcutInd*.

Shortcut: Shortcut Shortcut Shortcut Shortcut Shortcut Shortcut Shortcut Shortcut
Shortcut Shortcut Shortcut Shortcut Shortcut Shortcut

Text paragraph

- Bullet

Shortcut: ShortcutInd ShortcutInd ShortcutInd ShortcutInd ShortcutInd
ShortcutInd ShortcutInd ShortcutInd ShortcutInd

- bullet

Flag Paragraph Tags

The four flag tags in this section have been created for writers to use in internal drafts. They are based on the Warning paragraph tag from the standard *IPChap.doc* template.

There are four Flag paragraph tags. They are for internal use only and should be tagged with the Comment conditional text so that they don't appear in finished books. The online translator does not support these tags, so the bookbuild will break if you try to build a file without making the text conditional.

Examples of the four tags appear when viewing the templates through FrameMaker.

Following are examples of each of the four flag tags. You may want to copy an entire paragraph from this template, paste it into your document, and modify the text.

This example uses *FlagEditor*:



Flag for the Editor: Use this tag to bring something to the attention of your editor or production editor.

This example uses *FlagFiction*:



Fiction Alert: Use this tag to warn your reviewers about material to follow that may be inaccurate and needs close attention.



This example uses *FlagMissing*:

Missing Information: Use this tag to identify a place in the draft where you plan to add material later or need reviewers to supply input.



This example uses *FlagNew*:

New Information: Use this tag to alert your editor or reviewers that the following material is new to this revision of the document.

Advantages of Using the Flag Tags

Writers are encouraged to use these tags to communicate with their editors and reviewers, rather than inserting comments in brackets or in different fonts. The advantages of using these tags are as follows:

- They are easy for editors and reviewers to locate and distinguish from live text.
- They are easy to find when the document is final and it's time to remove them.
- They add visual interest and relieve the monotony.

Adding the Appropriate Icon to a Flag Paragraph

Each of the four flag tags has an icon in the left margin of the tagged paragraph. If you copy an existing flag paragraph from your document or from this template, you will copy the icon with it. In other cases, you must add the icon manually. Follow this procedure:

1. Place the cursor at the very beginning of a flag paragraph.

- Choose Special > Anchored Frame and select the correct settings according to the type of flag paragraph tag you are using. See Table 3-4.

Table 3-4 Anchored Frame Specs for Flag Tags

Paragraph Tag	Anchor Position	Baseline Offset	Near-Side Offset	Height	Width
<i>FlagEditor</i>	Left side	-0.4"	-1.5"	0.7"	0.6"
<i>FlagFiction</i>	Left side	-0.4"	-1.5"	0.7"	0.5"
<i>FlagMissing</i>	Left side	-0.4"	-1.5"	0.5"	0.7"
<i>FlagNew</i>	Left side	-0.2"	-1.5"	0.5"	1.0"

When you complete this and reactivate the page, you will see an empty box to the left of the flag paragraph.

- Choose View > Reference Pages and scroll to the Warning reference page. Copy the icon from the appropriate box. The icon is clearly labeled and has an explanation with it. After you copy it, choose View > Body Pages.
- Select the anchored frame you created and use the Edit > Paste command to paste the icon into the frame. *Do not* move the icon around in the frame. It is positioned correctly and moves with the text.

Managing Flag Paragraphs

Flag paragraphs are not supported in the translator, so they will not appear in the IRIS InSight versions of your documents. However, your document will build properly with flag paragraph tags in it if you mark them as conditional text. This must be done manually.

When it is time to remove the flag paragraphs from your document, you can find them quickly by using the FrameMaker Edit > Find/Change function:

1. Choose Edit > Find/Change.
2. Change the Find popup menu to Paragraph Tag.
3. Click the *Use Wildcards* button.
4. Type **f1ag*** in the text area.
5. Click *Find*.

Footnote Paragraph Tag

This is an example sentence that has a footnote at the end.¹

Page Break Paragraph Tag

The paragraph tag called *PageBreak* positions the paragraph that follows the tag at the top of the next page. To use *PageBreak*, create an empty paragraph immediately preceding the paragraph you wish to break to the next page, and apply the tag to the empty paragraph.

Character Tags

Character tags may appear within any paragraph tag. The only restriction on their use is that they should be applied only to actual visible characters or to spaces within a character-formatted string.

¹This is a footnote.

Many of the character tags in the character catalog are used to format non-editable text in such paragraph tags as Warning and FigTitle. Table 3-5 describes the character tags you may use for character formatting, along with their formatting specs.

Table 3-5 Character Formats

Tag	Print Format	Online Format Using SGIDOC DTD	Online Format Using SGIDOCBK DTD	Comments
Bold	bold	bold	bold	
Button	<i>italic</i>	<i>italic</i>	<i>body font</i>	For onscreen buttons only.
Callout	Helvetica 8p	NA	NA	For figures only, does not appear in main text
CalloutBold	Helvetica Bold 8pt Helvetica 8pt	NA	NA	For figures only; does not appear in main text
CalloutSmall	Helvetica 7pt Helvetica 8pt	NA	NA	For figures only; does not appear in main text
CmdLineOpt	Palatino bold	body-font bold	Courier	
Command	<i>italic</i>	<i>italic</i>	<i>Courier</i>	
DocTitle	<i>italic</i>	<i>italic</i>	<i>italic</i>	
FileName	<i>Palatino Italic</i>	<i>italic</i>	<i>Courier</i>	
Function	Palatino bold	body-font bold	Courier	
GlossaryItem	<i>italic</i>	<i>blue italic</i>	<i>blue italic</i>	See "Creating Glossary Links" on page 84.
HardwareLabel	Helvetica Bold 9pt	Helvetica Bold	Courier	
Italics	<i>italic</i>	<i>italic</i>	<i>bold</i>	
Keyword	Palatino bold	body-font bold	Courier	
Launchword	Palatino Regular 10pt; default text	red	red	Used for link launch in IRIS InSight
Refpage	Palatino Regular 10pt; default text	red	red	Used for reference page (man page) launch in IRIS InSight
ScreenDisplay	Courier 9pt	Courier	Courier	

Table 3-5 (continued) Character Formats

Tag	Print Format	Online Format Using SGIDOC DTD	Online Format Using SGIDOCBK DTD	Comments
Subscript	subscript	subscript	subscript	
Superscript	superscript	superscript	superscript	
Symbol	Σμμβολ	Σμμβολ	Σμμβολ	
UserInput	Courier Bold 9pt	Courier Bold	Courier Bold	
Variable	<i>Palatino Italic</i>	<i>Italic</i>	<i>Italic</i>	

Cross References

The IPTemplates include a standard set of cross-reference formats. You may use these or create your own, up to a maximum of twenty-four. If you need more custom formats, you may delete unused existing tags. However, do not change the specifications for any existing tags, because your modifications will go away if you or the next author ever reapply the template defaults.

The online tools do not recognize cross-reference format tags. They rely solely on the cross-reference string that appears in the text. This is why you can add any new formats you wish.

“Creating Online Links” on page 77 describes the IPTemplate cross-reference formats and how to use them.

Markers

There are five custom markers:

15 Marker #15 is used to override DTD restrictions on certain constructs: type "override" into the marker.

16 CrossBook Link is used to flag cross-book links for QA purposes.

17 Executable is used to execute external applications online.

If you want an icon in the margin, use this syntax:

```
ebt_launch:app:parms
```

If you want clickable text, use this:

```
Launchword:app:parms
```

If you want to launch Netscape and point your browser to a URL, use this:

```
Launchword:/usr/sbin/nr:<URL>
```

Use the second and third options in conjunction with the *Launchword* character tag to create a hot word, or set of words, that launches an executable. The marker should immediately precede the tagged word(s).

This paragraph contains a marker using *ebt_launch*. Note the icon that appears in the margin.

This paragraph contains hot text that launches an application.

See "Creating Links to URLs and Applications" on page 90 for complete instructions on how to use Marker Type 17.

18 The build tools translator turns this construct into a cross-book link, same as what you get with the standard FrameMaker cross-reference tool:

```
<marker#18>text<marker#19>
```

Marker #18 should have text using this syntax:

```
<target_book_short_title><space><target_element_unique_id>
```

19 Marker #19 ends a marker-created crossbook link (See marker #18 above). It is empty.

See "Creating Cross-Reference Links to Other Books" on page 82 for complete instructions on how to use Marker Types 18 and 19.

Conditional Text

The IPtemplates support six text conditions:

- Comment
- HelpSubTopic (see “Tagging FrameMaker Files for Help” on page 163 for details)
- IndexEntry (see “Using IndexEntry Conditional Text” on page 85 for details)
- OnlineOnly
- PrintOnly
- ColorPrintOnly

For information on how to use the FrameMaker conditional text features, see the FrameMaker documentation.

InSight Inline Object Tags

The IRIS InSight inline widget allows you to use digital media in your online document. The syntax is as follows:

type:referenced_filename

Movie_Title

For the first paragraph, use the *InlineObj* paragraph tag. *type* can be SGIVIDEO, SGIRGB, SGIAUDIO, or SGIINVENTOR. The title paragraph is optional. It uses the *InlineTitle* paragraph tag. Both *InlineObj* and *InlineTitle* must use the *OnlineOnly* conditional tag. For more information, see “Creating Inline Media Links” on page 90.

Help Paragraph Tags

Use the *HelpTopic* paragraph tag for embedding help in an online document. For instructions on creating a help document or embedding help in an online book, see Chapter 8, “Creating Online Help.”

Creating a Book File

You can create a book file as soon as you have at least one standard text file that will go into it. If you do so at this point, you will be able to take advantage of such book file utilities as automatic pagination and creation of generated files. You will also be able to open your files by double-clicking their names in the book file. Once you have created your book file, keep it open while working on any files it contains. This will help you keep track of your files and allow you immediate access to bookfile utilities.

To create a book file, follow these steps:

1. Open a document that will be part of your book.
2. Choose File > Generate/Book.
3. Choose the radio button that says "New multi-file book, including *filename.doc*."

A new window pops up on your screen within a few seconds. The header bar should have a name like *filename.book* and the main window should contain the name of the document you had open when you chose Generate. This new window is your bookfile.

After you have generated a new multi-file book, save the file as your bookfile using a prefix that will easily identify your book, such as *Prog.book*. This prefix will become the prefix for generated filenames. FrameMaker names generated files according to the title of the bookfile in which they reside. So, if your bookfile is named *Prog.book*, then you will have names like *Prog.TOC*, *Prog.LOT*, and so on. You cannot change these generated filenames later.

Added Versus Generated Files

"Add" and "generate" are two separate procedures in FrameMaker. "Add" means add a file pointer to a bookfile. "Generate" refers to the actual process of searching constituent files for designated paragraph tags and creating a "generated" file. Please note that the bookfile itself is also considered a "generated" file.

Adding Text Files to Your Book File

To add names of files created through standard text entry (chapters, appendixes, and so on) to your bookfile, follow these steps;

1. Choose File > Add File. The window that comes up has three main areas. The top portion allows you to choose the type of file you wish to add.
2. Choose *Document File*. The right field lets you choose where you would like the file to go in your bookfile. You can highlight any file in the list and chose Before or After Current File from the pulldown menu directly above the list. The left field lists the files you can add.
3. Choose the file you want from the left field and click *Add*. The name of the file should appear in the bookfile.
4. Highlight the new filename by clicking it in the bookfile, and go to File > Set Up File. The window that appears allows you to set the specifications necessary for your file to have the correct page and paragraph numbering. Table 3-6, Table 3-7, Table 3-8, and Table 3-9 contain guidelines on how to set these specifications according to the combination of part tabs and tabs included in your book.

Table 3-6 shows the specifications that should be set for text files that you add to your book when there are no part tabs or chapter tabs:

Table 3-6 Page Setup for Text Files (No Part Tabs or Chapter Tabs)

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix
Front.doc	right	restart	restart	not used
Intro.doc	right	continue	restart	not used
Chap1.doc	right	restart	restart	not used
Chap2.doc	right	continue	continue	not used
AppendixA.doc	right	continue	restart	not used
AppendixB.doc	right	continue	continue	not used
Glossary.doc	right	continue	continue	not used

Table 3-7 shows the specifications that should be set for text files that you add to your book when there are chapter tabs.

Table 3-7 Page Setup for Text Files (With Chapter Tabs)

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix
Front.doc	right	restart	restart	not used
Intro.doc	right	continue	restart	not used
Tab1.doc	right	restart	restart	not used
Chap1.doc	right	continue	continue	not used
Tab2.doc	right	continue	continue	not used
Chap2.doc	right	continue	continue	not used
TabA.doc	right	continue	restart	not used
AppendixA.doc	right	continue	continue	not used
TabB.doc	right	continue	continue	not used
AppendixB.doc	right	continue	continue	not used
Glossary.doc	right	continue	continue	not used

Table 3-8 describes the specifications that should be set for text files that you add to your book when there are part tabs.

Table 3-8 Page Setup for Text Files (With Part Tabs)

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix
Front.doc	right	restart	restart	not used
Intro.doc	right	continue	restart	not used
PartTab1.doc	right	restart	restart	not used
Chap1.doc	right	continue	continue	not used
PartTab2.doc	right	continue	continue	not used

Table 3-8 (continued) Page Setup for Text Files (With Part Tabs)

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix
Chap2.doc	right	continue	continue	not used
AppendixA.doc	right	continue	restart	not used
AppendixB.doc	right	continue	continue	not used
Glossary.doc	right	continue	continue	not used

Table 3-9 describes the specifications that should be set for text files that you add to your book when there are part tabs and chapter tabs.

Table 3-9 Page Setup for Text Files (With Part Tabs and Chapter Tabs)

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix
Front.doc	right	restart	restart	not used
Intro.doc	right	continue	restart	not used
PartTab1.doc	right	restart	restart	not used
Tab1.doc	right	continue	continue	not used
Chap1.doc	right	continue	continue	not used
PartTab2.doc	right	continue	continue	not used
Tab2.doc	right	continue	continue	not used
Chap2.doc	right	continue	continue	not used
TabA.doc	right	continue	restart	not used
AppendixA.doc	right	continue	continue	not used
AppendixB.doc ^a	right	continue	continue	not used
Glossary.doc	right	continue	continue	not used

a. Note that each appendix does not have its own chapter tab. Use just one chapter tab for all appendices in your book.

Adding Generated Files to Your Book File

To add names of files to be created through the FrameMaker file generation utility (Contents, Figures, Tables, Index), follow these steps:

1. Choose File > Add File.
2. From the top portion of the window that appears, choose the type of file you wish to add.
 - If it is a TOC, LOE, LOF, or LOT, choose Generated List, and use the popup menu directly to the right to select the type of generated file you wish to add.
 - If it is an Index, choose Generated Index and use the popup menu directly to the right to select Standard Index.
3. From the right portion of the Add File window, choose where you would like the file to go in your book file. You can highlight any file in the list and chose Add File Before or Add File After from the popup menu directly above the list. When you are satisfied with your choices, click *Add*.
4. The Set Up File window immediately appears. It provides various FrameMaker utilities with the information necessary to create and paginate your new generated file. Table 3-10 describes the settings you should use for each type of generated file:

Table 3-10 Page Setup for Generated Files

Filename	First Page Side	Page Numbering	Paragraph Numbering	Prefix	Include Paragraph Tagged
TOC.doc	right	continue	restart	not used	ChapTitle, GlossaryTitle, Heading1, Heading2, Heading3 (optional)
LOE.doc	right	continue	restart	not used	CodeTitle ExampleTitle
LOF.doc	right	continue	restart	not used	FigTitle
LOT.doc	right	continue	restart	not used	TableTitle
IX.doc	right	continue	continue	not used	Index

5. Once you have made your choices, press Enter and the new filename appears in your book file. You can tell it is a generated file by the plus symbol that appears immediately to the right of the name.

At this point, the file itself does not actually exist. What you have done is add the name of a file that FrameMaker will create when you choose Generate/Update from the File menu. See the section “Generating the TOC, LOE, LOF, LOT and Index” for instructions on how to generate these files.

Rearranging and Checking the Setup for Your Files

FrameMaker assumes that when you finally generate your generated files (as opposed to just adding their names to the book file), all the appropriate files are in your book file in the correct order, and have been set up with the correct specifications using File > Set Up File.

To rearrange files so they are in the proper order, use File > Rearrange Files. The window that appears is self-explanatory. This window is the only place where files can be deleted from your book.

After rearranging your files, use File > Set Up File to check for the proper specs, or fix any files that weren't set up when they were added to the book file. See Table 3-7 through Table 3-10 for the proper specs.

Generating the TOC, LOE, LOF, LOT and Index

The best way to create your generated files is as follows:

1. Set up your book file as described in “Creating a Book File” on page 65.
2. Copy the corresponding IPTemplates for your generated file to your working directory.
3. Rename the copied template files using the same names that these sections have in the book file.
4. Select File > Generate/Update.

FrameMaker assumes that it is overwriting existing files and simply dumps the generated text into a pre-formatted template section. You should then be able to open up your generated file and see it already fully formatted.

One way to tell if this procedure hasn't worked is if unformatted files pop up at the end of the generation process. This indicates that FrameMaker did not recognize the renamed template copies you put in with your document files, probably because you misnamed them. If this occurs, simply apply the templates to your generated files by using File > Import > Formats. Then choose File > Generate/Update to regenerate the content.

Troubleshooting Autonumbering Problems

The template uses a single autonumbering scheme to run Chap/AppNum, the "#-" in the footer, Headings, Tables, Figures, Examples, and Equations. The scheme relies upon your having built a book, set up your files, and used File > Generate/Update to repaginate the files.

If your autonumbering doesn't work, here are some possible causes:

- You haven't built the book, set up your files in your book, or repaginated using File > Generate/Update.
- You've "hardwired" the autonumbering.
- One or more chapters or appendices don't have the proper *Num* or *Title* tag.
- You're using an old or hybrid version of the template.

Using FrameMaker Files and Template Features

This chapter provides information on how best to use FrameMaker features to produce online books, as well as pointers to related information elsewhere in the book. This chapter contains these sections:

- “Tagging FrameMaker Files” provides some important rules for using paragraph and character tags in your FrameMaker files.
- “Creating Online Links” explains the different types of links and how they appear and behave in the IRIS InSight viewer. This section also provides guidelines and instructions for creating links, such as internal, cross-book, glossary, index, reference page, and launchable applications, in your book.

For information on creating and importing figures, see the instructions in Chapter 5, “Working With Figures.”

For instructions on creating online help and integrating it into a GUI application, see Chapter 8, “Creating Online Help.”

Tagging FrameMaker Files

Before you begin, be sure you are using the standard SGI IPTemplate. The example files that come with the templates contain information on basic tagging sequences and standard uses of the various tags.

This section offers some general tips for tagging.

Using Character Tags, Not Character Formats

If you want individual characters to have a particular look, be sure to use the supplied character tags to change their appearance. Do not use FrameMaker's character designer tool to simply change the format of a character. The build tools look only at tags, not at any modifications you may have made to a character's format. If you don't understand the distinction between tags and formats, refer to the FrameMaker documentation.

Using Valid Tag Sequences

The example files that come with the templates and Chapter 3, "Using the IPTemplates to Author Documents" contain descriptions of the various tags and where and how they may be used. For example, don't put the ChapTitle tag before the ChapNum tag, or the FigTitle tag before the ChapTitle tag, or the Heading2 tag directly after the ChapTitle tag. The online translator expects to see only certain sequences of paragraph and character tags and produces errors for what it considers invalid sequences.

Note: Don't create your own tags, because the build tools cannot translate any tags they don't know about.

Using the Correct Character Tag

Use the correct character tag for each tagged word or phrase. Content-based tags (based on the kind of information being tagged) are preferable to format-based tags (based on the physical appearance of characters). For example, if you're tagging a command, use the Command character tag rather than the Italics character tag. Use the format-based character tags, such as Italics and Bold, only when the template doesn't contain a specific tag that fits your word or phrase.

Creating Equations as Figures

The current build tools do not support FrameMaker equations. If you want an equation in your online book, you must use *snapshot* or some other screen capture tools to capture a picture of the equation. *snapshot* saves the equation as a *.rgb* file and you can import this *.rgb* file as if it were a figure. Don't give the equation a figure title, though, or it will show up in the list of figures as a figure. For more information about creating equations, see "Equation Paragraph Tags" on page 43; information about screen captures can be found in "Capturing Images" on page 99.

Using Character Tags on Tabs, Soft Returns, or Hard Returns

In general, try to keep character tags on the specific word or phrase you're tagging, without tagging surrounding whitespace. In particular, never apply a character tag to a tab or a soft return¹. It's easy to apply character tags to whitespace characters by mistake, particularly when you're working with a hanging list (note that a tab character that has a character tag generates a warning message), so be careful.

Here are a couple of things to watch for when tagging:

- If possible, select the word you're tagging by double-clicking on it. This keeps you from accidentally selecting extra spaces or tabs. (Unfortunately, sometimes double-clicking does not select the entire "word" you're tagging, because the word is a C expression or something else containing unusual characters.)
- Sometimes it is difficult to see your selection clearly in FrameMaker, so it's hard to tell if you've selected spaces or tabs in addition to words. If you've accidentally tagged a hard return or a soft return along with the preceding word, insert a space between the return and the word before it. Tag this space with the Default Paragraph Format character tag. This procedure is necessary because hard and soft returns in FrameMaker take the character tag of the character preceding them; you can't just tag the return itself to fix the problem.

Using Tabs in Code

Format code examples with spaces, rather than with tabs. Tabs in code don't display properly online because the translator doesn't recognize them for that paragraph tag. Conversion scripts exist to convert tabs to spaces before importing code examples into FrameMaker files.

Note: Be sure to turn off FrameMaker's Format > Document > Text Options > Smart Spaces feature before you import any code examples into your FrameMaker file. Otherwise, multiple spaces will be squashed down to one space and you'll lose your formatting. (Also turn off Smart Quotes so that code samples use straight quotation marks instead of curved ones.) Be sure to turn on Smart Spaces (and Smart Quotes) after you're done working with code samples.

¹You can create a soft return by holding down the Shift key and pressing Enter.

Creating Valid Copyright and Trademark Symbols

Create copyright, registration, and trademark symbols according to the instructions listed here. If you use any other method for creating these symbols, you'll run into problems when you convert your files to SGML.

- To create the copyright symbol (©), hold down Ctrl and press Q then release both keys, hold down Shift and type a right parenthesis ()).
- To create a registered trademark symbol (®), hold down Ctrl and press Q, then release both keys, hold down Shift, and type a left parenthesis ((). Apply the Superscript character tag to the registered trademark symbol after it is inserted. If you don't know whether to use a trademark symbol or a registered-trademark symbol, consult your company's legal department.
- To create a trademark symbol (™), hold down Ctrl and press Q, then release both keys, hold down Shift, and press asterisk (*). Apply the Superscript character tag to the trademark symbol after it is inserted.

Creating Valid Dashes

Create en dashes and em dashes according to the instructions listed here. If you use any other method for creating these symbols, you might create problems when you convert your files to SGML.

- To create an en dash (–), hold down Ctrl and press Q, then release both keys, hold down Shift, and press P.
- To create an em dash (—), hold down Ctrl and press Q, then release both keys, hold down Shift, and press Q.

Creating Footnotes

Footnotes and table footnotes are supported in IRIS InSight. To create them, use the Special > Footnote feature and the Footnote paragraph tag (for footnotes within text) or the TableFootnote paragraph tag (for footnotes in tables).

Creating Online Links

Links are connections between items of information, whether within the same online book or between separate online documents. When the reader clicks a linked item (whether text or an icon), either the IRIS InSight viewer scrolls to the referenced material (a target), or the material appears in a popup window (depending on what kind of material is referenced).

Links fall into two groups based on whether they are automatically generated from the structure of the document (structural links) and require no writer effort, or explicitly created by the writer during document development (writer-generated links).

About Automatic Links

A structural link is automatically created from the structure of the document during the conversion process. Lists of such links appear in the IRIS InSight > Views menu. All the headings listed in the Table of Contents (TOC) view in InSight, for example, are automatically linked to the corresponding headings within the book itself. The same goes for all figures, tables, and media shown in their respective View lists. The result is that a reader can click any item in a list view and the text area scrolls to the appropriate location in the online book.

About Writer-Generated Links

The writer can use various FrameMaker tools to create links to text, tables, or figures in the same book (or in another book), between the text and the glossary, or between the index and the text. Each link has its own appearance and behavior.

Table 4-1 summarizes the writer-generated link types:

Table 4-1 Cross-Reference Link Types

Link Type	How to Create	Appearance in InSight	Link Behavior
Cross-reference to text or table within a book	Use the Special > Cross-Reference format	Cross-reference appears in bold blue in the text	Viewer scrolls to referenced information
Cross-reference to a figure within a book	Use the Special > Cross-Reference format	Cross-reference appears in bold blue in the text	Viewer scrolls to referenced figure
Cross-reference to another book	Use the Special > Marker > Types 18 and 19 as described in "Creating Cross-Reference Links to Other Books" on page 82	Cross-reference appears in red in text	Referenced information appears in new viewer
Glossary link	Use the FrameMaker character tag, GlossaryItem; then create a glossary entry in your book's glossary file as described in "Creating Glossary Links" on page 84	Item appears in blue italicized text	Definition appears in a pop-up window
Index link	Use the Special > Marker > Index marker as described in "Creating Index Entries" on page 84	Index entry appears in the Index View. Instead of a page number, location in the text appears as bold blue cross- reference to the associated heading/title	Viewer scrolls to referenced information
Reference page link	Tag an item with the RefPage character tag as described in "Creating Reference (Man) Page Links" on page 89	Item appears as red text	Text window appears containing indicated reference page

Table 4-1 (continued) Cross-Reference Link Types

Link Type	How to Create	Appearance in InSight	Link Behavior
Inline media	Use the InlineObj and InlineTitle paragraph tags as described in “Creating Inline Media Links” on page 90	Object appears in media viewer inside InSight viewer window	Media viewer controls appropriate to media type allow user to manipulate or view media
Launch an application (including a World Wide Web browser)	Use Special > Marker > Type 17, as described in “Creating Links to URLs and Applications” on page 90	Either icon in margin or hot text, depending on how you set up the link	Application comes up in a new window

A Word on Cross Referencing Headings

It is recommended that for each chapter and appendix of your book, you begin with a bulleted list of the top-level sections in that part, making part of each item a cross-reference to the specified heading. This approach serves two purposes:

- It allows the online reader to click on any item in the list and have the viewer scroll to that point in the text.
- It creates cross-reference targets on all top-level headings. Whenever you create a cross-reference, FrameMaker automatically creates a target marker at the referenced location. These targets can be used by other writers to link to your book (see “Creating Cross-Reference Links to Other Books” on page 82).

Creating Cross-Reference Links Within a Book

Cross-reference links within a book are created using the FrameMaker Cross-Reference tool. They appear as bold blue text within the online document. When the reader clicks the link, IRIS InSight either scrolls to the text if the link is to a table or text, or opens the figure in a restricted view if the link is to a figure.

In the example shown in Figure 4-1, the cross reference is referring to Figure A, and Figure A is located two paragraphs below the cross reference. When the reader clicks on the blue “Figure A” text, they are automatically jumped to the image and title for Figure A.

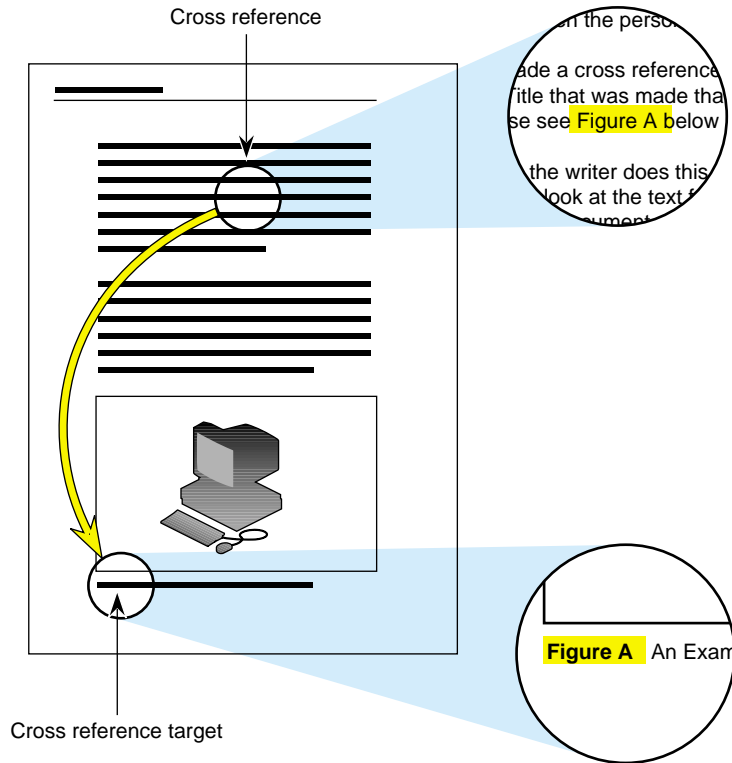


Figure 4-1 Diagram of Cross Reference Targets

To establish a cross-reference:

1. Insert your cursor where you want the link to appear.
2. Choose **Special > Cross-Reference**.
The Cross-Reference dialog box appears.
3. Select the paragraph tag of the item you to which you want to link from the list of paragraph tags in the left field.
4. Select the appropriate heading, title, or text in the right field.
5. Choose the appropriate cross-reference format from the Format popup menu.
6. Click *Insert*.

Table 4-2 shows a matrix of the possible cross-reference formats for each tag.

Note: Only paragraph tags and their possible cross-reference formats that can generate a useful cross-reference are listed in Table 4-2.

Table 4-2 Cross-Reference Format by Paragraph Tag

Paragraph Tag	Cross-Reference Formats Supported by SGIDOC DTD	Cross-Reference Formats Supported by SGIDOCBK DTD ^a
ChapTitle	AppRef, AppRef1st w/com, AppRef1st w/o end punct, AppRef1st w/per, ChapRef, ChapRef1st w/com, ChapRef1st w/o end punct, ChapRef1st w/per	AppRef, ChapRef, ChapRef1st w/o end punct
CodeTitle	FgTblExEqRef	FgTblExEqRef
ExampleTitle	FgTblExEqRef	FgTblExEqRef
FigTitle	FgTblExEqRef	FgTblExEqRef
Heading*	HdRef, HdRef & App, HdRef & Chap, HdRef & Page, HdRef w/comma, HdRef w/period	HdRef, HdRef & App, HdRef & Chap, HdRef & Page
InlineTitle	FgTblExEqRef	FgTblExEqRef
List1st	ParaNumber	ParaNumber
List*	ParaNumber	ParaNumber
TableTitle	FgTblExEqRef	FgTblExEqRef

a. The cross-reference formats containing embedded punctuation (w/com, w/per) may still be used in documents built with the SGIDOCBK DTD, however the punctuation will be lost in the translation.

Be sure to use an appropriate cross-reference format. Note that the conversion tool automatically removes any page numbers that appear in your cross-reference text; if your book will appear in print as well as online, it's a good idea to use cross-reference formats that include page numbers to help hard copy readers navigate through the book.

Creating Cross-Reference Links to Other Books

Cross-book links appear as bold red text within the online document. When the reader clicks on the link, IRIS InSight opens the appropriate book (if it's available) and either scrolls to the text if the link is to a table or text, or opens the figure in a restricted view if the link is to a figure.

Although cross-references between documents are much like cross-references between files within a book, there are two problems that complicate the situation tremendously:

- There is no way to guarantee that any two books (even those documenting the same product) will be revised simultaneously, so it's possible that another book you link to may change after your book releases, breaking your link to the other book.
- You usually don't have the same access to another book that you have to your own; in particular, you can't create a new link target in the referenced book.

The first problem is a coordination and document management issue. If you want to create cross-book links, please take the time to evaluate the risk of having it break. If the book you want to link to is in development, be sure to talk with the author. If the book has already been released, find out when it will be revised and how extensive the changes will be.

Once you have decided to create a cross-book link, you must make sure that there's a cross-reference target in the other book at the point to which you want to link. If such a target doesn't exist already, you need to ask the writer of the other book to create a target for you. It's probably not a good idea to make changes to someone else's book without asking first, especially if you use a revision control system to check documents into and out of a central archive.

To create a cross-book link using markers:

1. Open the FrameMaker chapter in your book and insert the cursor at the beginning of the text where the cross-book link marker will be placed.
2. Go to the FrameMaker menu bar and choose Special > Marker.
3. In the Marker dialog, select Type 18 from the Marker Type pop-up menu.

4. In the Marker Text box, enter information about the cross-book link using this syntax:

target_short_title<Space>*unique_id*

where *target_short_title* is the short title of the target book, and *unique_id* is the identifying number associated with the target element in the target book. To identify the correct *unique_id*, look in the SGML file for the target book; find the target element and look for an XREF LINKEND= tag nearby. For instance, in the following example the *unique_id* for the TITLE element is 57332:

```
<SECTION ID="LE57332-PARENT"><TITLE ID=LE57332-TITLE">Supplementary
Reading</TITLE><PARAGRAPH>Refer to these documents to supplement the
information in this guide:</PARAGRAPH>
```

Or, open the FrameMaker file containing your target information and find the marker associated with the title or heading you to which you want to link. Highlight the target marker and note the marker ID number shown in the Marker Text field. This is the *unique_ID* you want to use in your cross-book link.

5. Press the *New Marker* button to create a new marker containing the information you've specified.
6. Move the cursor from the beginning of the text where the cross-book link will be placed to the end of the text.
7. From the Marker dialog box, select Type 19 from the Marker Type popup menu. Leave the Marker Text box empty.
8. Press the *New Marker* button to create a new empty marker of type 19.
9. If the cross-book link is the title of another book, tag the entire construct (markers and the title in between them) with the DocTitle character tag.

The result is a piece of your document with this structure:

```
<type-18_marker>text<type-19_marker>
```

where "text" is the text that will be marked as a link, the type-18 marker contains the name of the target book and the ID of the target element, and the type-19 marker is empty (a placeholder to indicate the end of the cross-reference text).

Note: The text between the markers cannot contain any character tags other than DocTitle.

Creating Glossary Links

Glossary links appear as blue italicized words in the online text. When the reader clicks on the word, the definition appears in a restricted viewer. The link itself is either to the *SGI Glossary of Terms*, a global glossary common to all books, or to a local glossary specific to the book. IRIS InSight searches the local glossary first (if it exists), then searches the global glossary.

To make a glossary item, tag the word using the `GlossaryItem` character tag. Do this only where you think it is important, not every time the word appears; otherwise your book will be peppered with blue italicized glossary links. In order to match, the text that you tag `GlossaryItem` must be spelled and punctuated identically to the text of a `GlossaryTerm` in your glossary.

Note: This restriction (requiring the linked reference to be identical to the entry) can cause some problems. For instance, if you have a glossary entry for the term “window,” you can’t link to it from the words “windows” or “windowing” in the text; and it can be difficult to re-word the text to fit the format of the glossary entry. One workaround for this problem is to create additional cross-referencing glossary entries for those words—have an entry for “windows,” for instance, which simply says “see ‘window,’” with the word “window” linked to the glossary entry. (You can put glossary links inside the glossary itself.)

To create a local glossary tailored specifically to your book, use the `IPGlossary.doc` template provided with the book template files. Once you’ve created your glossary, add its filename (usually simply *glossary.doc*) to your *Makefile* at the end of the list of chapter files. The build tools automatically create links between words tagged `GlossaryItem` and the corresponding entries in the glossary file.

Creating Index Entries

Index entries appear in the Index View in much the same way as they would in a hard copy index. The major difference is that instead of a page number, the location in the text appears as a bold blue cross-reference to the associated heading or title.

Use FrameMaker index markers to create index entries for the back-of-the-book index (so called to distinguish it from the full-text index that the build tools create).

Using Index Markers

To create index entries using markers:

1. Insert the cursor next to the text to be indexed.
2. Choose Special > Marker.
3. Select Index from the Marker Type popup list.
4. Type the index information in the Marker Text field.

For second- and third-level index entries, a colon is used as the delimiter. Follow the syntax

`<main_entry>: <sub-level>`

5. Click *New Marker*.

To edit an existing index entry:

1. Choose Special > Marker.
2. Highlight the index marker so the text appears in the Marker Text field.
3. Edit the text.
4. Click *Edit Marker*.

Using IndexEntry Conditional Text

As an alternative, if you are trying to update a large set of index markers, use the conditional text format, IndexEntry. The IndexEntry conditional text format is part of the standard set of conditional text formats provided with each template file.

To determine if your document includes index entries created using this conditional text technique:

1. Choose Special > Conditional Text.
2. Click *Show/Hide*.
3. Highlight IndexEntry and use the <-- button to move it to the left window.
4. Click *Set*.

To find an existing index entry done with IndexEntry conditional text:

1. Choose Special > Marker to display the Marker dialog box.
2. Choose Edit > Find/Change.
3. Set up Find to Marker of Type: and type **Index** into the dialog box.
4. In the Change popup menu, select To Text.
5. Click *Find*.

When you have found the index entry you want to edit:

1. Click *Change* so the index marker information changes to text.
2. Edit the index entry in the text of the document (it will appear in blue underlined text).
3. Select the index entry text (highlight it).
4. Ensure that the new text appears in the Marker Text field.
5. Click *Edit Marker*.

To hide conditional text index entries in a document:

1. Choose Special > Conditional Text.
2. Select *Show/Hide*.
3. Select IndexEntry and move it to the Hide column using the --> button.
4. Select *Set*.

To hide conditional text index entries in an entire document.

1. Open a document in a book that has its index entries hidden.
2. In the bookfile, Choose File > Import Formats.
3. From the *Import from Document* scroll list, select the document that has its index entries hidden.
4. Select only the *Conditional Text* check box.
5. Ensure that all files you wish to update are listed in the Update scroll list.
6. Click *Import*.

Indexing Tips

Below are some tips to help make index creation a less painful experience:

- Be careful about how you treat blank spaces before and after your index entries. For example, if you put an index entry at the beginning of a paragraph and put an unconditional blank between the index entry and the real beginning of the text, your paragraph will be improperly indented.
- FrameMaker can automatically generate lists of markers. In the bookfile, choose File > Generate/Book, check *List*, select List of Markers or Alphabetical Marker List, and click *Generate*.
- In your FrameMaker files, choose Edit > Find/Change to search for Any Marker, Marker of Type:, or Marker Text.

Online books require certain limitations in index entries:

- Index entries can be no more than three levels deep.
- You can't put an index marker in a table footnote.
- A character format in an index entry can't cross a semicolon or colon boundary; for instance, `<Italics>entry:subentry<Default Para Font>` won't work the way you might expect it to. Instead, you'd have to use `<Italics>entry<Default Para Font>:<Italics>subentry<Default Para Font>`.
- Special characters that result in *entity* references (such as special characters for foreign languages) may not sort correctly in the online index.
- Sorting ignores these characters if they appear as the first character in an index entry: `"`, `<`, `$`, `.`, `/`, and `<space>`. The syntax for overriding the ignore instructions are outlined in "Template-Specific Index Entry Syntax Rules" on page 88.

Template-Specific Index Entry Syntax Rules

For an explanation of the standard rules for index entry syntax, see the FrameMaker documentation. The templates have defined the following characters as “ignored” for index sorting purposes:

```
$ dollar sign  
/ slash  
. period  
< less than  
" double quote  
" left double quote  
" right double quote  
- hyphen  
- endash  
- emdash
```

So, for instance, the entry

```
/etc/hosts
```

is sorted as though the slashes were absent. The full entry (under E) appears as

```
E  
/etc/hosts
```

Be aware that because *all* instances of an ignored character are ignored, sequences like the following are possible:

```
/etc/gghosts  
/etchosts  
/etc/root
```

This is primarily a problem with the / character, since the other ignored characters rarely appear anywhere but the beginning of an index entry. In the case of the / character, it was decided that the advantage of having the leading / ignored outweighed the disadvantage of having to put up with a few cases where an ignored / in the midst of an entry caused an unusual sort order.

The following rules apply to ignored characters in index entry syntax.

- If an ignored character appears in brackets, it is “unignored” for sorting purposes:

`/etc/hosts[/]` is sorted (under `Symbols`) as:

```
Symbols
/etc/hosts
```

- If the ignored character in brackets is followed by any text, the entry is sorted according to the full bracketed string, and the ignored character *is* ignored.

`/etc[/etc]` is sorted as:

```
E
/etc
```

- The unignore “flag” can be grouped with other sort instructions in brackets.

`/etc[; / ;blah]` is sorted as:

```
Symbols
/etc
```

and

```
B
/etc
```

and

```
E
/etc
```

Creating Reference (Man) Page Links

To create a link to a reference page, use the `Refpage` character tag. Be sure to tag the entire reference page name, including the parentheses, if there are any. The reference page name will appear in red text online and, when clicked, will launch the reference page in a shell. In print, the reference page name appears in the default font (Palatino).

Creating Inline Media Links

IRIS InSight supports the inclusion of digital media files created from SGIVIDEO, SGIRGB, SGIAUDIO, or SGIINVENTOR. To create a link to digital media in your online document, perform the following steps:

1. Open the FrameMaker file and insert your cursor where you want the inline media to appear in your document.
2. Press Enter to create an empty paragraph tag.
3. Tag the paragraph with the InlineObj paragraph tag.
4. Type the inline object information following the syntax:

type:referenced_filename

where type can be SGIVIDEO, SGIRGB, SGIAUDIO, or SGIINVENTOR.

5. Press Enter. The new paragraph is automatically tagged as InlineTitle.
6. Type the title of your media.
Note: The title paragraph is optional. If you don't need a title for your media to appear in the online document, delete this paragraph.
7. Highlight and tag both the InlineObj and InlineTitle paragraphs with the OnlineOnly conditional tag.

An example of an inline media link can be found here in the online version of this book.

Here is an example of an SGIINVENTOR file.

SGIINVENTOR:test.iv

An Example of an SGI Inventor File

Creating Links to URLs and Applications

IRIS InSight supports the ability to launch a Web browser and point it to a particular Universal Resource Locator (URL) as well as to launch an application using the Launchword character tag in conjunction with Marker Type 17.

Creating a Link to a URL Address

To launch Netscape and link to a URL address, perform the following steps:

1. Open the FrameMaker file and insert your cursor where you want the link to appear in your document.
2. If a URL is very long, put it on a line by itself by either pressing Enter to create a new paragraph or by pressing Shift +Enter to create a soft return.
3. Type the URL.
If the URL is longer than one line, make it wrap after a slash.
4. Insert the cursor in front of the URL.
5. Choose Special > Marker.
6. Select custom marker 17 from the Marker Type scroll list.
7. In the Marker Text box, type
Launchword: /usr/sbin/nr:<URL>
8. Click *New Marker*.
9. Tag the URL with the Launchword character tag, but do not tag the marker.

The URL will appear as default font (Palatino) in print and as red text in your online document.

Creating a Link to an Application

To launch an application from your online book, perform the following steps:

1. Open the FrameMaker file and insert your cursor in front of the name of the application.
2. Choose Special > Marker.
3. Select custom marker 17 from the Marker Type scroll list.
4. In the Marker Text box, type
Launchword: /<path>/<to>/<program>
For example, to launch the application MovieMaker, type
Launchword: /usr/sbin/moviemaker
5. Click *New Marker*.

6. Tag the application name with the Launchword character tag, but do not tag the marker.

The application name will appear as default font (Palatino) in print and as red text in your online document.

Working With Figures

The online build tools can support six image formats for use in figures. An extension (suffix) on the file name of the imported image designates the format of the image. You can have figures in your book that don't use the supported image formats, including composite figures composed of a combination of image formats, by means of a workaround. Except for FrameMaker line art, all images must be imported by reference into a FrameMaker document, from the *print* subdirectory of your book directory.

Original image files and the components of composite figures are stored in a directory called *orig*. During the build process, the images in *orig* are processed and placed into two additional figure-related directories that the build process generates: *print*, which contains processed images for the hard copy book; and *online*, which contains processed images for the online book. All image files in *orig* must be listed in the *Makefile* to specify how the build tools should process them.

This chapter explains how to prepare figures during the book development process. The chapter contains these sections:

- “Supported Image Formats”
- “Image File Naming Conventions”
- “The Figure Subdirectories”
- “Adding Image Files to the Makefile”
- “About FrameMaker Art”
- “Capturing Images”
- “Processing and Importing Image Files”
- “Using Unsupported Image Types”
- “Using Custom print Files”
- “Things to Remember About Figures”

Supported Image Formats

The bookbuilding tools currently support the image formats listed in Table 5-1. Note in Table 5-1 that composite figures (figures containing multiple imported images in a single anchored frame) are supported for some, but not all, image formats.

Table 5-1 Image Formats Supported by the Build Tools

Image Format	Number of Files per Anchored Frame
RGB	One or more
FrameMaker line art	One or more
RGB and FrameMaker line art	One or more of either
Encapsulated PostScript, including Adobe Illustrator	One
GIF	One
XWD	One

Unsupported composite formats, such as figures composed of multiple Adobe Illustrator files or figures composed of multiple file formats (a combination of RGB and PostScript, for example), can be included in a book. However, these composites require a workaround (see “Using Unsupported Image Types” on page 104).

All image files, including the elements of composites, must be stored in the *orig* directory. As with supported figures, the elements of unsupported composites must be listed in the *Makefile* and imported from the *print* directory.

Image File Naming Conventions

The *orig* directory normally contains at least one image file for each figure in your book (composite figures may use more than one file). The filename for each image file in the *orig* directory must end with a standard extension (or suffix) that designates the format of the image.

Table 5-2 shows the standard extension for each image format that is currently supported. Notice that Table 5-2 does not include an extension for FrameMaker line art, since line art figures are not imported files.

Table 5-2 Image File Naming Conventions

File Format	Extension
Adobe Illustrator Encapsulated PostScript	.ai
color RGB	.rgb
black and white RGB	.bw ^a
Non-Illustrator Encapsulated PostScript	.eps, .ps or .PS
GIF	.gif
XWD	.xwd

a. These images are screen snapshots (or other RGB files) that have been converted to black and white with the *tobw* utility.

The Figure Subdirectories

The local working directory for a book contains three subdirectories for use with figures:

orig Place all original image files and the components of figure composites in this directory. All RGB snaps, Adobe Illustrator EPS files, and other original pieces of art belong in this directory.

Hint: For books that are printed in color, store the color versions for the printed copy in *orig*; import a black and white FPO (For Position Only) version into the FrameMaker document from the *print* directory.

print This directory, once you've populated it by using the *make _print* command, contains all image files that will be used in the printed version of your book. Any image files that are imported into anchored frames must be imported from the *print* directory. Normally there's no reason to alter *print* or its contents through any means other than the *make _print* command.

online This directory, once you've populated it by using the *make _online* command, contains all image files that will be used in the online version of your book. Normally there's no reason to alter *online* or its contents through any means other than the *make _online* command.

Adding Image Files to the Makefile

The *Makefile* controls the conversion of images into the appropriate format for the online and printed versions of your book. During the build process, the build tools automatically convert imported file references into the appropriate references to the *online* directory.

The *Makefile* contains variables that specify image formats and conversion processes. Use the following instructions when listing image files in *Makefile*. List each image file in only one place in the *Makefile*. You might also find it useful to look ahead to Table 5-3 when deciding where to list image files in the *Makefile*.

A completed description of the Makefile and how to edit all its parameters can be found in "Editing the Makefile" on page 110.

PRINT_BW = Use this variable to specify original RGB color images that will be imported as black and white images for the printed book. The files that you list on the PRINT_BW line appear as eight-bit color GIF images in the online book. All image files listed on the PRINT_BW line must be stored in the *orig* directory with a *.rgb* extension.

Note: Files listed in the PRINT_BW variable can't be part of a composite figure.

PRINT_COLOR = Use this variable to specify RGB color images that will appear in color in both the printed version and the online version of your book. The files that you list on the PRINT_COLOR line appear as color GIF images in the online book. All image files listed on the PRINT_COLOR line must be stored in the *orig* directory with a *.rgb* extension.

Note: If you want images to appear in 24-bit color online, do not list them on this line; list 24-bit color images on the RGB variable line instead.

- EPS = Use this variable to specify all Encapsulated PostScript images (including Adobe Illustrator EPS files) in your book. These images appear in color in FrameMaker and in the online version of your book; they appear in color in the printed version if printed on a color printer. Image files listed on the PostScript line must be stored in the *orig* directory with either a *.ps* or a *.ai* extension.
- GIF= Use this variable to specify all GIF images in your book. All image files listed on the GIF line must be stored in the *orig* directory with a *.gif* extension. Color GIF files appear in color in the online version of your book; black and white GIF files appear in black and white in the online version of your book. Be sure to use GIF 87a format instead of GIF 89a; you can use the *file* command to find out what format a given file is in. Some version of the *togif* utility translate to GIF 89a; be sure to use the version of *togif* that comes with InSight Professional Publisher.
- RGB= Use this variable to specify images that should be processed for 24-bit color in the online version of a book. All image files on the RGB line must be stored in *orig* with an *.rgb* extension. Note that 24-bit images may appear dithered on some monitors, and IRIS InSight doesn't display in 24 bits by default; many or most customers won't see your images in 24 bits even if you include 24-bit images with your book.
- Note:** RGB files appear in color in the FrameMaker application; this contributes to printing problems due to file size. Also note that using this variable greatly increases the size of an online book, because RGB-format images are much larger than GIF-format images. Unless you're certain that you need 24-bit images in your book, it's best not to use this variable.
- XWD= Use this variable to specify all XWD images in your book. All image files listed on the XWD line must be stored in the *orig* directory with a *.xwd* extension. Color XWD files will appear in color in both the printed and the online versions of your book; black and white XWD files will appear in black and white in both the printed and the online versions of your book.
- BW= Specifies RGB-format images that have already been converted to black-and-white format.

About FrameMaker Art

The drawing tools in FrameMaker allow users to work with straight and curved lines, circles, polygons, and text. Frame artwork is PostScript-based and commonly used for simple illustrations or callouts to imported artwork.

To create FrameMaker line art:

1. Press Enter to create a new paragraph.
2. Apply the *Fig* paragraph tag to the paragraph.
3. Press Enter again to create a paragraph that automatically has the *FigTitle* paragraph tag applied to it. Type in the figure title.
4. Place the cursor in the *Fig* paragraph and Choose Special > Anchored Frame.
5. In the Anchored Frame dialog box, choose these settings:
 - Anchoring Position: Below Current Line
 - Alignment: Right
 - Cropped
 - Size: width = 5.375" and height = your choice (the frame can be resized once it's inserted)
6. Click *New Frame*.
7. Choose Graphics > Tools.
8. Use the drawing tools with a line weight of 0.5 pts. to create your simple diagram.
9. Use the A button to create a text cursor and click it in the anchored frame.
10. Choose Callout from the font catalog and type the callout.
 - Avoid using CalloutBold and CalloutSmall.
 - Do not use text boxes.
11. Select the callout using the Object Selection tool (the black arrow in the Tools GUI) and position the callout using the Ctrl and arrow keys.

Capturing Images

It's often necessary to use screen captures as illustrations in books. There are a variety of tools available to aid you in producing screen captures; use whichever seems most appropriate to your needs.

Tools and Where to Get Them

snapshot

The *snapshot* utility (*/usr/sbin/snapshot*) is part of the IRIX 6.2 and later system software, in the *eo.e.sw.gltools* software subsystem. *snapshot* reads an area of the screen specified by the user and saves it to a file named *snap.rgb* in your home directory. This utility can capture an entire screen. For more information on *snapshot*, refer to the *snapshot* man page.

bsnap, ssnap, and winsnap

IRIS InSight Professional Publisher includes three shell-script capture utilities, all of which can be found in */usr/share/Insight/bin* and are part of the *insight_dev.sw.tools* subsystem. All three scripts require */usr/sbin/scrsave* (which is part of the *eo.e.sw.gltools* subsystem in the IRIX 6.2 and later distributions).

To use these tools, you must append */usr/share/Insight/bin* to your path or PATH environment variable. This variable is initialized in your *~/.cshrc* file if you use the C shell or a shell derived from the C shell, and in your *~/.profile* file if you use the Bourne shell or a shell derived from it. Be sure to use the *rehash* command after editing your *.cshrc* file.

<i>bsnap</i>	Snaps a bordered image without rubber-banding and captures screen images to RGB. Allows variable time delay with the <i>-s</i> command-line option. Includes window borders by default; to avoid window borders, use the <i>-n</i> option.
<i>ssnap</i>	identical to <i>bsnap</i> , except that after capturing a window, it shrinks the window to 65% of its size in both horizontal and vertical directions, then brightens and sharpens the result. <i>ssnap</i> requires <i>/usr/sbin/izoom</i> and <i>/usr/sbin/hipass3</i> , both of which are part of the <i>eo.e.sw.imagetools</i> subsystem.
<i>winsnap</i>	Has no command-line options. Doesn't capture window borders.

capture

In addition to these tools, if you have an Indy or O2 system you probably have the `/usr/sbin/capture` utility installed on it. `capture`, an interactive program that records digital media files, is part of the `dmedia_tools.sw.movietools` subsystem. It can capture images from an IndyCam or O2CAM, a video input, or the screen. See the `capture(1)` reference page for more information.

Screen Capture File Names

Name your screen capture figure files descriptively, so that someone else can tell which figures go where. The filenames should end in the `.rgb` suffix. For example, if your book is called *MyBook*, and you have three figures in chapter one, you might name the figures `mybook_fig1.1.rgb`, `mybook_fig1.2.rgb`, and `mybook_fig1.3.rgb`. Or you might use descriptive names such as `01.1.sample.rgb`, `01.2.menubar.rgb`, and `01.3.dialog.rgb`.

Screen Capture Procedure

The screen capture procedure is slightly different depending on what you want to capture: full windows, portions of windows, or pull-down menus. Note that for any of these approaches, if you want to show your screen capture against a white background, you must first place the background, then place the window you're capturing on top of that background. One way of doing this is to open a shell window with a white background:

```
% xwsh -bg white
```

Screen Captures of Windows

If you want to capture an entire window, with or without its borders, use `bsnap`.

1. In a shell window, change to the directory where you are going to store the screen captures (the `orig` subdirectory of your book's working directory).
2. Decide what you're going to name the screen capture. (See "Screen Capture File Names" on page 100 for some guidelines.)
3. Bring the window you want to capture to the front, leaving part of your shell window visible.

4. If you want to include the window's border (including its title bar), launch *bsnap* from the shell window:
% **bsnap** *filename*
5. If you don't want to include the window's border, launch *bsnap* like this:
% **bsnap -n** *filename*
6. Click anywhere in the window you wish to capture.
7. Wait a few seconds for the capture to complete.
8. Verify that the capture worked as desired:
% **ipaste** *filename*
9. If the image isn't as you expected, repeat steps 3 through 8.

Screen Captures of Portions of Windows

If you only want to snap part of a window, it's best to use *snapshot*.

1. Change to the directory where you are going to store the screen captures (the *orig* subdirectory of your books' working directory).
2. Decide what you're going to name the screen capture. (See "Screen Capture File Names" on page 100 for some guidelines.)
3. Bring the window you want to capture to the front, leaving part of your shell window visible.
4. Start *snapshot* from the shell window:
% **snapshot**
5. Use the right mouse button in the *snapshot* window to bring up the *snapshot* menu. Select "New file name" from that menu. Enter the name you want your screen capture to have. Press Enter.
6. Move the *snapshot* window out of the way:
 - Move the cursor over the *snapshot* window.
 - Press Alt+F7.
 - Move the cursor; the *snapshot* window's outline moves with it.
 - Click the left mouse button when the window is where you want it.

7. With the cursor still in the *snapshot* window, hold down the Shift key to let *snapshot* retain keyboard and mouse focus.
8. Still holding down the Shift key, move the camera cursor into the window you want to capture.
9. Drag out a rectangle, using the left mouse button, around the portion you want to capture.
10. Move the cursor off to one side and, still keeping the Shift key depressed, use the right mouse button to bring up the *snapshot* menu. Choose "Save as *filename*" from that menu.
11. To make additional screen captures, repeat steps 7 through 10.

Screen Captures of Pull-Down or Popup Menus

Use the same procedure for pull-down menus as for complete windows, except:

- Use *bsnap -s 5* to make *bsnap* delay five seconds before beginning the capture.
- After you click in the window you're capturing from, quickly pull down the menu you want to capture. Keep it pulled down for several seconds, until *bsnap* finishes.

Processing and Importing Image Files

All image files used in a book must be imported into a FrameMaker document from the *print* directory for a book to build successfully.

This procedure explains how to generate and populate the *print* and *online* directories and how to import images into FrameMaker files from *print*:

1. If you've already copied the *Makefile* template to your local working directory, skip to step 2. If you haven't already done this, do it now:

```
% cp /usr/share/Insight/templates/make/Makefile_sgidocbk Makefile
```

(for books to be built using the SGIDOCBK DTD)

or

```
% cp /usr/share/Insight/templates/make/Makefile_sgml doc Makefile
```

(for books to be built using the SGIDOC DTD)

Note: The functionality of the two DTDs is essentially the same; the online appearance is different, however, especially for character tags. Before converting all your bookbuilding to the new SGIDOCBK DTD, you'll want to perform a sample bookbuild and view it in IRIS InSight to ensure its online appearance more closely matches your company's style guide than the appearance created by SGIDOC.

The *Makefile* file specifies information about your book to the build tools. It contains variables that you must set and instructions for how to set them. See "Editing the *Makefile*" on page 110 for additional information.

2. Change write permissions on your copy of the *Makefile* file.

```
% chmod 664 Makefile
```

3. Edit the *Makefile* template.

```
% vi Makefile
```

(You can use a different text editor, such as *emacs* or *nedit*, if you prefer. If you use FrameMaker to edit this file, be sure you save as text when you're done editing.) Set the image variables in the *Makefile* that apply to your imported image files, such as PostScript and PRINT_BW (see "Processing and Importing Image Files" on page 103). At this point you can also set the other variables, such as TITLE and BOOK_FILES, if you wish, but only the image variables are required to generate the *print* directory.

4. Execute the *make _print* command:

```
% make _print
```

The *make _print* command creates a *print* directory and copies the files from *orig* to *print*, converting any formats as needed.

5. Import your image files as described in “Figure Tags” on page 46. See Table 5-3 for filename extensions.

Open each FrameMaker file that should contain imported images. The method that you use to import the image files depends on the current importing settings in the document files:

- If your FrameMaker files do not specify *orig* or *print* as the location of imported image files, the FrameMaker application warns that it can't import the files. To continue, specify the *print* directory as the new location of your image files.
- If your FrameMaker files specify *orig* as the location of the imported image files, the FrameMaker application does not issue a warning; it opens the files, importing images from the *orig* directory. In this case, you must re-import all of the image files, specifying the *print* directory as the image file location. To re-import image files, temporarily rename the *orig* directory, then reopen the document file to generate a FrameMaker error. Specify the *print* directory version of the image files in the error dialog box. When the re-importing is complete, rename the temporary directory to *orig*.

Using Unsupported Image Types

If your book contains images with unsupported formats, use this workaround to prepare the images for processing by the bookbuilding tools. This procedure assumes that you already have an anchored frame in your FrameMaker file that contains an unsupported image format:

1. Snap the composite picture as an RGB file.
2. Copy the new RGB file to the *orig* directory.
3. Add the new image file to the PRINT_BW line in *Makefile*.
4. Execute the *make _print* command to add the file to the *print* directory.

5. Create a second frame in your FrameMaker document:

Using the same Fig paragraph tag, insert a second anchored frame below the location of the original frame containing the unsupported image.

Note: Be sure that you have a single figure title for the anchored frame pair. The title should appear below the second frame.

6. Import the *.bw* image into the new frame from the *print* directory.
7. Tag the frame containing the unsupported version of the figure with the PrintOnly conditional text tag.

The PrintOnly tag appears on the Conditional Text menu.

8. Tag the frame containing the *.bw* version of the figure with the OnlineOnly conditional text tag.

The OnlineOnly tag appears on the Conditional Text menu.

Note: When you print the document, be sure to turn off the OnlineOnly condition using the Show/Hide selection on the Conditional Text menu. During the bookbuilding process, the PrintOnly condition (among others) is turned off automatically.

Using Custom *print* Files

Occasionally, you may find that the quality of a printed version of an image created by automatic conversion is unacceptable. This may happen, for instance, if your images have a black background that shows up fine online, but overwhelms any reversed-out line art in print. To produce a custom version for print:

1. Perform a screen snap on the original version and put it in the *orig* directory.
2. Edit the *Makefile* as usual and execute *make _print*.
3. Modify a copy of the original file and put it in the *print* directory.

This modification might involve either of these tasks:

- Running an image processing tool on the file.
 - Changing the X resource values for an application to increase contrast, and re-snapping a window with contrasts acceptable for a black and white book.
4. Import the *print* version of the figure as usual.

Note: This process works because *make* does not reprocess an image file for the *print* directory if the file in *print* is more recent than its counterpart in the *orig* directory. For this reason, you *must* recreate a custom print version whenever you update the original. If you do not recreate the custom version, *make* will replace the custom version with the newer, automatically generated version that does not contain your changes.

Things to Remember About Figures

For your book to build properly (without errors) you must follow all the rules in this list:

- Import all images *by reference* from the *print* directory. When you build your book, the online tools automatically grab the corresponding images from the *online* directory for display in IRIS InSight.
- Import each image into an anchored frame. For figures that appear within normal page boundaries, the frame must be anchored to an empty paragraph (blank line) that is tagged with the Fig paragraph tag.
- In order for a figure title to appear in the “List of Figures” in IRIS InSight, the Fig paragraph tag has to be followed by FigTitle paragraph tag.

Table 5-3 lists the various file formats that can occur in the *orig* directory. For each format, it shows the format of the file in a FrameMaker document, which produces the hard copy version of the figure, and the format of the file in IRIS InSight, which produces the figure in online books. Table 5-3 also shows the file extension in the *orig* and *print* directories and the *Makefile* variable where the file should be listed.

Table 5-3 Image Formats for Hard and Soft Copy

File Format in <i>orig</i>	FrameMaker File Format	IRIS InSight File Format	Suffix in <i>orig</i>	Makefile Variable	Suffix in <i>print</i>
RGB	black/white RGB	GIF	.rgb	PRINT_BW	.bw
black/white RGB	black/white RGB	GIF	.bw	BW	.bw
RGB	RGB	GIF	.rgb	PRINT_COLOR	.clr
RGB	RGB	24-bit RGB	.rgb	RGB	.rgb
Adobe Illustrator Encapsulated PostScript	Adobe Illustrator Encapsulated PostScript	GIF	.ai	EPS	.ai

Table 5-3 (continued) Image Formats for Hard and Soft Copy

File Format in <i>orig</i>	FrameMaker File Format	IRIS InSight File Format	Suffix in <i>orig</i>	<i>Makefile</i> Variable	Suffix in <i>print</i>
Encapsulated PostScript	Encapsulated PostScript	GIF	<i>.eps or .ps</i>	EPS	<i>.eps or .ps</i>
GIF	GIF	GIF	<i>.gif</i>	<i>GIF</i>	<i>.gif</i>
XWD	XWD	GIF	<i>.xwd</i>	<i>XWD</i>	<i>.xwd</i>

For all image formats except black/white RGB, color palette information remains unchanged during image file processing. This means that a black and white original produces a black and white figure, and a color original produces a color figure.

For the black/white RGB format, color palette information is reduced to black and white during processing. This means that both black and white and color originals produce black and white figures.

Building Online Books

The bookbuilding process creates SGML files from FrameMaker source files. It also generates the additional files that are required to view the SGML using the IRIS InSight viewer or the SGI Help viewer. The build process also generates error reports to help you debug source files and create error-free online documents.

To invoke the build process, use the *make* command. You can give *make* a variety of arguments to perform different build-related tasks. The *make* command reads a file called *Makefile*, which describes the book and the source documents using a set of variables. Once a book is built, writers and production editors can proofread it while viewing it in IRIS InSight, and/or test the resulting online help.

Note: The 4.2tools are backward-compatible, so books undergoing revision can use the new *Makefile_sgidocbk*, which uses the SGIDOCBK DTD (commondocdefs) or the older *Makefile_sgml doc*, which uses the SGIDOC DTD (commondocrules). The functionality of the two DTDs is essentially the same; the online appearance is different, however, especially for character tags. Before converting all your bookbuilding to the new SGIDOCBK DTD, you'll want to perform a sample bookbuild and view it in IRIS InSight to ensure its online appearance more closely matches your company's style guide than the appearance created by SGIDOC.

This chapter explains how to build a book, generate error reports, and view the built book with IRIS InSight. The chapter includes these sections:

- "Editing the Makefile"
- "Files and Directories Generated by make"
- "Arguments to the make Command"
- "Using make Commands"
- "If make Doesn't Work"
- "make book Error Messages"
- "Viewing a Built Book"

Note: The build process is more efficient if you compose and import figures, and then use the File > Generate/Update command in the FrameMaker book file, *before* beginning the build.

Editing the Makefile

If you don't have a *Makefile* in your book's working directory, follow the instructions in "Makefile" on page 22.

The *Makefile* specifies the FrameMaker files that compose the book and gives information about the figure files in the book. The *Makefile* also specifies the short title and full title of your book, the part number and date, any foreign language attributes, and the bookshelf where the book is displayed. An edited version of the *Makefile* that contains the specifications for your book must be stored in your local working directory to build the book.

Note: Be prepared to supply the short title (see "Choosing a Book's Short Title" on page 21), bookshelf, and a list of your document and figures files during this procedure, though you can change any of these things and rebuild the book later, if necessary.

The *Makefile* is self-documenting; it contains instructions on what to add and where to add it. The procedure that follows provides supplementary information for editing the *Makefile*.

1. Edit the TITLE variable.

Specify the short title for your book as the value for the TITLE variable. For help choosing a short title, see "Choosing a Book's Short Title" on page 21.

Note: Do not use the short title of your book as the basis for the name of any files in your document other than the book file. For example, if *short_title.book* is the name of your book file, then *short_title.doc* is not a valid filename (though *short_titleCh1.doc* is a valid filename).

2. Edit the FULL_TITLE variable.

Specify the full title for your book as it appears on the cover. For help choosing a full title, see "Choosing a Book's Title" on page 20.

3. Edit the VERSION variable.

Specify the version number, month and year for this revision. The version number for an SGI book is the part number for the book. For example:

```
007-2863-001, 9/99
```

4. Edit the BOOKSHELF variable.

Specify the directory in which your company's bookshelves are located and bookshelf on which your book is to appear in the IRIS InSight library, in this form:

```
/usr/share/Insight/library/companyname_bookshelves/companyname
```

(You can use a standard abbreviation of your company's name as your *companyname*, but all books from a given company must use the same abbreviation.) For instance, a company named YoyoDyne Systems would specify

```
/usr/share/Insight/library/YoyoDyne_bookshelves/YoyoDyne
```

for the BOOKSHELF variable; the books would then appear in IRIS InSight under a bookshelf named "YoyoDyne." Note that since *companyname* is part of a directory name, it must follow all rules for UNIX filenames—for instance, *companyname* can't contain characters such as spaces, slashes, or asterisks.

SGI publications can appear on any of three bookshelves: SGI_EndUser, SGI_Developer, and SGI_Admin. Field Engineer manuals go on the SGI_Service bookshelf. All SGI bookshelves are located in the */usr/share/Insight/library/SGI_bookshelves* directory.

5. Edit the BOOK_ICON variable.

Specify which icon should represent your book on the bookshelf in IRIS InSight. Valid icon choices for SGI publications are *sgi_end_user*, *sgi_developer*, and *sgi_admin*. SGI Field Engineer manuals use the "binder" icon. Books developed outside of SGI use the "generic" icon, which is the default.

6. Edit the BOOK_FILES variable:

Specify the source files that compose your book, in the order in which they appear in the book. Include the front matter, introduction, chapter, appendix, and glossary files in the order in which they appear in the book. Do not include any generated files (index, TOC, LOE, LOF, or LOT). Also leave out chapter tabs.

Table 6-1 lists the FrameMaker files that you should and should not include in the BOOK_FILES variable.

Table 6-1 FrameMaker Files to Include and Omit in Makefile

Include	Omit
Front matter file	Table of Contents
About This Guide or Introduction file	List of Figures
Part tabs	List of Tables
Chapter files	List of Examples
Appendix files	Index
Glossary file	Book file
	Chapter tabs

7. Specify figure information:

Use the information in “Adding Image Files to the Makefile” on page 96 to complete this step. The *Makefile* file contains several variables that you can edit to specify how to process your figures files to display them in the online version of your book. When you assign values to figure variables, keep these facts in mind:

- The *make* commands can only process the original figure files in the *orig* directory. This processing populates the *print* and *online* directories with correctly formatted figures.
- The *print* directory is generated by the *make _print* command; it contains figure files that are processed specifically for the paper version of a book. All figure files that are imported into FrameMaker files must be imported from *print*—so if you create a new figure in the *orig* directory, you must use *make _print* to put a usable version of that figure in the *print* directory before you can import the figure (from the *print* directory) into your book.
- The *online* directory is generated by the *make _online* or *make book.full* command; it contains figure files that are processed specifically for the online version of a book.

8. Edit the INLINE_MEDIA variable, if needed.

Specify the name of inline media objects for movie, audio, or Inventor files if your book contains any.

Files and Directories Generated by *make*

You can run *make* with a variety of different arguments. The argument that you use determines the generated files that result from the build. The *make* command produces four different types of files:

- files that are actually part of the online book
- image files for use in figures in the printed version of the book
- intermediate files
- error files

Intermediate files are artifacts of the build process. *make* must produce these files in the course of creating an online book, and the files can be used to speed up subsequent book builds when only part of the book source has changed. The intermediate files are also used to create error files, which are reports of various types of source errors detected during the build process. Error files are not necessary steps in the build process and, though it is generally not recommended, it is possible to create an online book without generating any error files.

Note: For instructions on how to find and fix errors reported in the various error files, see Chapter 7, “Finding and Fixing Online Build Errors.”

Since the build process is a sequence of steps, *make* commands that take source files closer to a finished online book necessarily require that all earlier steps are completed. Likewise, generating an error file that relies on certain intermediate files will cause those intermediate files to be updated, if necessary, before an error report is generated.

A complete bookbuild can take anywhere from 15 seconds to 30 minutes or more, depending on the size of the book, the speed of your workstation, and the amount of activity on your workstation. Subsequent builds take less time, depending on the number of source files that you edited since your previous build.

Figure 6-1 shows where the various intermediate and error files are produced during the build process.

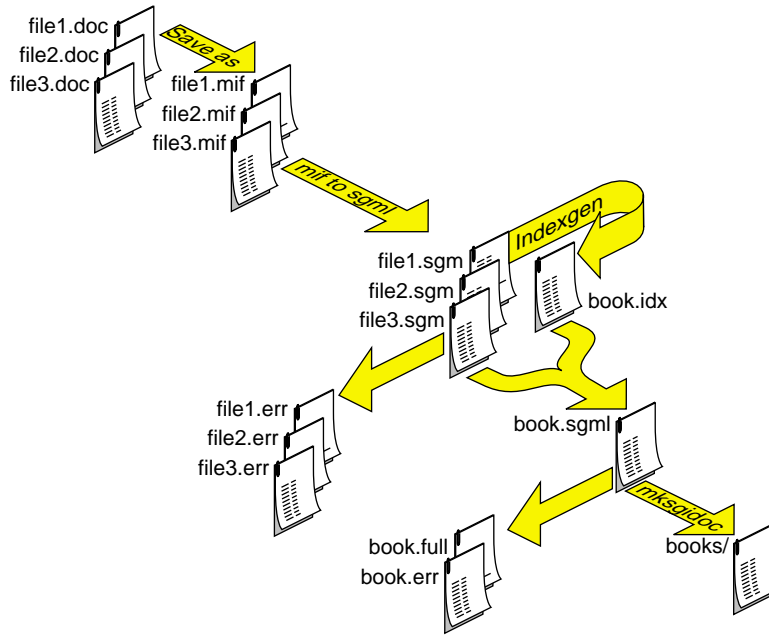


Figure 6-1 Files Produced During the make Process

Intermediate Files

The *make* command generates a sequence of intermediate files in the process of creating an online document. Some are descendants of individual source files, while others combine information generated from all the source files for a book. All intermediate files appear in your document directory. They are generated automatically as part of the online build process. By using different *make* commands, you can produce a single file or set of files instead of all the generated files.

- .mif* files These are ASCII versions of the FrameMaker *.doc* or *.fm* files, in Frame’s Maker Interchange Format. The *mif2sgml* translation software reads and converts these files to SGML.
- .sgm* files These are individual SGML files, produced from the *.mif* files by the *mif2sgml* translation software. One *.sgm* file is generated for each FrameMaker file listed in the *Makefile*.

- booklist.txt* file This file contains the bookshelf, book icon, language, title, short title, part number, and hidden variable information from the *Makefile*.
- short_title.idx* file
This file contains index information for the SGML version of the book.
- short_title.sgml* file
This file contains the concatenated SGML for all the files listed in the *Makefile*, plus all the information from the *short_title.idx* file.
- short_title.full* file
This file contains the concatenated *.err* files, the portion of the Makefile containing the variables (title, part number, bookshelf, book icon, etc.),

Error Files

Error files list errors found in the various intermediate files. For instructions on how to find and fix errors, see Chapter 7, “Finding and Fixing Online Build Errors.” For information on error files, see “Error Files” on page 128.

- .err* files This file contains errors and warnings generated from the corresponding *.sgm* or *.sgml* file. You should fix all messages labeled *ERROR*, but you may ignore the messages labeled *WARNING*.
- short_title.full* file
The `make book.full` command generates this file. It contains the concatenated *.err* files, the portion of the Makefile containing the variables (title, part number, bookshelf, book icon, etc.), a report on unresolved cross-references, a list of multiple link targets, and a list of unresolved glossary links.

The *books* Directory

The `make book` command produces a subdirectory called *books* in your working directory that contains the built version of the book. See “Viewing a Built Book” on page 124 for more information.

Arguments to the *make* Command

The *make* command takes many arguments, giving you flexibility to build different pieces of your book or selected error files. Since the build process is a sequence of steps, *make* commands that require more steps necessarily take longer. Figure 6-1 shows the basic sequence.

Note: See “Summary of Software/Configuration Requirements” on page 9 for a list of subsystems that must be installed on your workstation prior to using the *make* command.

Table 6-2 lists the commands that operate on all files in a book. To perform any action listed in the table, enter the command exactly as shown.

Table 6-2 *make* Commands For the Entire Book

Command	Output
make book	A <i>books</i> subdirectory containing a viewable book.
make book.err	<i>short_title.err</i> file, containing all ERROR and WARNING messages for the book.
make book.full	<i>short_title.full</i> file containing all error information, including link error reports and unresolved glossary terms; also a <i>books</i> subdirectory containing a viewable book.

Table 6-3 lists the commands for individual FrameMaker files.

Table 6-3 *make* Commands for Single FrameMaker Files

Command	Output
make <i>filename.mif</i>	FrameMaker MIF file from <i>filename.doc</i>
make <i>filename.sgm</i>	SGML file from <i>filename.mif</i>
make <i>filename.err</i>	error/warning report on <i>filename.sgm</i>

Table 6-4 lists the commands that operate on figure files. Note the underscores; *make* can get confused when its argument is both a build rule and a directory name, so you should use *make _print* and *make _online* when building figures instead of just *make print* and *make online*.

Table 6-4 *make* Commands for Figures

Command	Output
<i>make _print</i>	newly updated figure files in the <i>print</i> directory
<i>make _online</i>	newly updated figure files in the <i>online</i> directory (note that <i>composite figures</i> are not updated by this command; to update composite figures, you must rebuild the <i>.sgm</i> file that contains them)

Table 6-5 lists the commands that remove previously-built files. Note that none of these commands removes your source files (such as your FrameMaker document files and the files in the *orig* subdirectory). However, take care in using these commands if you've modified any generated files (such as image files in *print* or *online*) by hand.

Table 6-5 *make* Commands for Removing Files

Command	Effect
<i>make clean_book</i>	Removes the <i>books</i> subdirectory (which contains the online book, if previously built) and all generated MIF files, plus the <i>short_title.sgml</i> file
<i>make clean_reports</i>	Removes the error report files, including the <i>.full</i> file
<i>make clean</i>	Removes all of the above, as well as any temporary files that have been left in place
<i>make clean_figures</i>	Removes all files from the <i>online</i> subdirectory
<i>make clean_sgm</i>	Removes generated SGML files
<i>make clobber</i>	Removes all of the above
<i>make clean_print</i>	Removes files in <i>print</i> subdirectory
<i>make clobber_all</i>	Removes all of the above

Using *make* Commands

The *make* command argument that you choose depends on the stage of development of your book. Use these recommendations as a general strategy for choosing a *make* argument:

- Run *make book.full* on a book fairly early in the development process to generate error reports that you can use to begin debugging. Try to fix as many of these errors as possible before creating your first viewable book.
- After fixing most of the errors from your initial *make book.full*, you can produce *.err* files for individual source files, as needed, as you continue development. You should also build the entire book periodically using *make book* or *make book.full* to monitor the progress of your online book.
- Finally, before you incorporate the final book files into your software build, you must run *make book.full* and verify that your *short_title.full* file contains no errors. Then make sure the newly generated book is viewable in IRIS InSight, that all figures and tables work, and that the book looks the way you want it to look.

The *make* commands provide some information as they run; this information appears in the window from which you issued the command. If a *make* command encounters an unresolvable error, it reports an error message. If a *make* command terminates abnormally, see “If *make* Doesn’t Work” on page 121 for suggestions on finding and fixing the problem.

For instructions on how to find and fix errors that appear in the various error reports generated by *make*, see Chapter 7, “Finding and Fixing Online Build Errors.”

Initial *make*: *make book.full*

As a first step in the debugging process, use the *make book.full* command to build the SGML files with full error reports. (This command also generates a viewable copy of the book.) These reports give you an idea of how far along your book is, and how much work it may take to get it online. Try to eliminate as many of the easy, obvious errors as you can before you try and view your book. After you’ve fixed most of the errors, you can rebuild the viewable book using *make book* or *make book.full*.

To run *make book.full*, follow this procedure:

1. Close all your FrameMaker files. (The files must remain closed until the MIF files have been generated from them; you may reopen them at any point after MIF files are generated.)
2. Change directories to your local working directory.
3. Generate the SGML files and error reports.

```
% make book.full
```

This converts your FrameMaker files into SGML and creates a complete error report file that you can use to debug your book. (“Checking the .full File” on page 133 and “Finding and Fixing Errors” on page 142 explains how to do this debugging.) This complete error report is called *short_title.full*, where *short_title* is the *short title* of your book. The *make book.full* command also generates individual *.err* files, one for each chapter of the book; you can examine those files individually if you prefer, instead of the entire *.full* file.

make book.full usually takes from 30 seconds to 30 minutes depending on the size and contents of the book.

Debugging Individual Files: *filename.err*

There’s no need to rebuild the entire book every time you make a change. This takes time and CPU cycles, and in many cases the amount of noticeable change is fairly minimal. A better strategy is to clean up and rebuild files individually, rebuilding the entire book only when you feel you have accumulated significant changes.

You can produce an error report (a *.err* file) for either individual files or the complete book. The *.err* files list errors pertaining to document structure as well as information about incorrect tag use. To produce a *.err* file for a single source file, type:

```
% make filename.err
```

If you want to see a concatenated version of all your *.err* files, use this command:

```
% more *.err
```

To save the output in a file, add *> filename* to the end of your command.

To produce a *.err* file for an entire book, type:

```
% make book.err
```

See “Finding and Fixing Errors” on page 142 for instructions on fixing errors.

Making a Viewable Book: *make book*

If you want to view your book online and aren’t interested in the error reports, follow these steps:

1. Close all your FrameMaker files. (The files must remain closed until the MIF files have been generated from them; you may reopen them at any point after MIF files are generated.)
2. Change directories to your local working directory.
3. Generate the book.

```
% make book
```

This command builds the files needed to display your book in the IRIS InSight viewer. “Viewing a Built Book” on page 124 explains how to view the book in IRIS InSight.

Verifying an Error-Free Book: *make book.full*

Run *make book.full* to verify that you have cleaned out all the errors in your book and to make sure that your online book builds. Then check the newly built book with the IRIS InSight viewer to make sure it looks the way you want it to.

To build your final book:

1. Close all your FrameMaker files. (The files must remain closed until the MIF files have been generated from them; you may reopen them at any point after MIF files are generated.)
2. Change directories to your local working directory.
3. Type the *make* command.

```
% make book.full
```

4. Check the *short_title.full* file to make sure there are no errors or unresolved references or glossary terms.

5. To view the online book, type:

```
% iiv -b .
```

If *make* Doesn't Work

If a *make* command doesn't work, use these suggestions to correct the problem:

- Try to find and interpret the error message. Usually, the final cryptic "error code 2" (or similar error) will have been preceded by a more coherent message farther back in the build output. Look for phrases such as "cannot process", "cannot find", or "file not found."
- Make sure that you renamed *Makefile_sgidoc* or *Makefile_sgidocbk* to *Makefile*, and that the permissions are right: you need permission to read and write the *Makefile*. For information on setting up your *Makefile*, see "Copying Template Files to Your Working Directory" on page 22.
- Make sure all your FrameMaker files are closed. Look for stray *.lck* files in your working directory. These are "lock" files that should exist only as long as the corresponding Frame file is open. (Note that *.mif* files cannot be generated if the corresponding *.doc* or *.fm* files are in use.)
- Make sure all the filenames in the *Makefile* are spelled correctly, and that there are no extra spaces after the short title, or after any of the line continuation characters (`\`).
- Examine the *Makefile* to verify that lists of files extending over multiple lines have a backslash (`\`) at the end of every line except the last.
- Run *df* to make sure you have room in *TMPDIR* (type `echo $TMPDIR` to get the actual path; when *TMPDIR* is not defined, it defaults to `/usr/tmp`). The *make* commands (particularly the figure-processing parts) can use quite a bit of space in the temporary directory specified by the *TMPDIR* environment variable; if there's no room, the commands won't work properly. See the `tmpnam(3S)` man page for further information.
- Verify that your filenames (particularly figures) end in the appropriate extensions (suffixes).
- Remember to use `make _print` instead of `make print` (and `make _online` instead of `make online`).
- Try using `make -n book.full` (or use `-n` with any of the *make* commands). This tells you exactly what *make* is trying to do. Look for extra spaces, misspelled filenames, or anything else that doesn't look quite right.
- Try copying a new *Makefile_sgidocbk* or *Makefile_sgml doc* into your working directory from `/usr/share/Insight/templates/make`, and moving the entries from your current *Makefile* to the new one.

make book Error Messages

make book errors occur during execution of the *make book* or *make book.full* commands. This section contains a list of such error messages, their causes, and how to fix them.

Note that these errors are different from the errors that occur during translation from MIF to SGML. *make book* errors appear in a shell window while a *make* command is running, and usually prevent the command from completing until you fix the problem. Translation errors are listed in error files (such as the *.err* and *.full* files) and don't usually prevent a *make* command from completing. For information on translation errors, see "MIF-to-SGML Translation Errors" on page 143.

Problem: File Already Locked

```
FrameMaker reports error -1 "File already locked by login @ machine on date"  
*** Error code 1  
smake: 1 error
```

login is a login name (probably yours), *machine* is a machine name (probably yours), and *date* is a date (probably the current date). This means *make* can't open one (or more) of the FrameMaker files because it's locked. Close all your FrameMaker files (from within FrameMaker) and try again. If you don't have any open FrameMaker files, look through your book directory and remove any files that end in *.lck*. Try *make book* again. Another possible problem could be that someone else has logged in to your machine and moved or opened your files.

Problem: Can't Open File

```
====Building master SGML File====  
cat: cannot open .sgml.tmp  
*** Error code 2  
---Short_title ---  
cat: cannot open .sgml.tmp  
***Error code 2  
smake: 2 errors
```

Here, *Short_title* is the short title of your book. This error message isn't quite so informative, though. If you get something like this, open your *Makefile* and look at the line that lists the short title of the book. Look for extra spaces or punctuation. In the example shown above, there is an extra space following the short title of the book. Make sure the short title listed in the *Makefile* is the same as the name of your book file.

Problem: *commonbookdefs* File Errors

```
make: file '/usr/share/Insight/templates/make/commonbookdefs' line 3: Must be a separator (:  
or ::) for rules (bu39)  
make:file '/usr/share/Insight/templates/make/commonbookdefs' line 3: Syntax error
```

The *dev.sw.make* software subsystem is not properly installed. Re-install it.

```
"Makefile", line 16: Could not find /usr/share/Insight/templates/make/commonbookdefs  
"Makefile", line 51: Could not find include file '${COMMONBOOKDEFS}'  
Fatal errors encountered -- cannot continue
```

This error indicates that the build tools are not properly installed or mounted.

Problem: Can't Open *booklist.txt* File

```
mkbook warning: can't open file './booklist.txt'.
```

The *booklist.txt* isn't writable. Make it writable using *chmod*. The *booklist.txt* file is generated by the build tools; don't edit it by hand.

Problem: Must Have *impr_rip.sw.impr*

```
--- online/figurename.ai ---  
ps2gif: ERROR - you must have impr_rip.sw.impr for this to work
```

The specified figure is a Encapsulated PostScript file, but the tools you need to process PostScript files are not installed. Either install them (they're in the *impr_rip.sw.impr* software subsystem) or use only non-PostScript figures.

Problem: Cannot Map *libnsl.so* Library

```
29292://usr/frame/bin/sgi.irix.mips/fmbatch: rld: Fatal Error: cannot  
map soname 'libnsl.so' using any of the filenames  
/usr/lib/libnsl.so:/lib/libnsl.so:/lib/cmplrs/cc/libnsl.so:/usr/lib \  
/cmpls /cc/libnsl.so:/usr/lib/libns  
--- font.mif ---  
*** Error code 1  
---localfigrules ---  
*** Error code 208
```

This indicates that the networking software for the license manager isn't installed. Install the *coe.sw.svr4net* software subsystem.

Viewing a Built Book

It is important to view a built book regularly during the development process to evaluate the online presentation of your material. It might be useful to review the information provided in "Guidelines for Organizing an Online Book" on page 4, and "Guidelines for Cross-Referencing in Online Books" on page 6, in completing an evaluation of an online book.

Viewing an Uninstalled Book

Use the following command to display a book with IRIS InSight during the book development process. This command assumes that you're in your working directory.

```
% iiv -f -b .
```

In this command, the arguments to the *iiv* command (**-f -b .**) direct IRIS InSight to launch in the foreground (ignoring any other copy of IRIS InSight that might be running) and to look for a *books* directory in the current directory. You can also specify other pathnames (including relative pathnames) in an *iiv* command to open books in different directories. However, any directory that you specify must contain a local *books* directory that was created by a successful book build.

To view an uninstalled book on a different workstation, log in remotely to the other workstation, go to the appropriate directory, and launch IRIS InSight:

```
% rlogin guest@hostname  
% cd bookbuild_directory  
% iiv -f -b .
```

Viewing an Installed Book

In addition to viewing a book during the development process, you might also want to verify the installed image of your book. Viewing the installed image shows the final form of the online book as it will appear to the intended audience.

Note: Before using this procedure, be sure that your workstation contains the disk space that the book requires. If you are viewing help material, you will also need additional disk space for the application that your material documents.

Use this procedure to view an installable book:

1. Determine the location of the installable book images.

An installable image of a book is part of the software subsystem that the book documents. Find out from where to install the subsystem image. The software engineers for your product should be able to provide this information.

2. Use *swmgr* to install the book on your workstation.

Use the standard software installation process to install your book as described in “Installing Subsystems Using Software Manager” on page 14.

Note: You might be required to install prerequisite subsystems before installing a book. To install help material, for instance, you must also install the associated application.

3. Start IRIS InSight to view the book.

```
% iiv
```

This command launches IRIS InSight and displays the newly installed book on a bookshelf.

4. Click the book’s icon or title to open the book.

Note: If you are viewing help material, start the associated application and use the application’s help feature to view your material.

5. Launch InfoSearch to view the book.

```
% infosearch
```

This command launches the InfoSearch browser.

6. Click Online Books to display the bookshelves.

7. Click the bookshelf on which your book is installed.

8. Click the book’s folder icon or title to open the book.

Finding and Fixing Online Build Errors

This chapter describes the errors and warnings that can occur when FrameMaker files are converted to SGML format using the *mif2sgidocbk* translator.¹ It compares the FrameMaker and SGML versions of a sample file and identifies the intermediate files that are created to store errors in the translation process. It also lists some common error messages and explains how to correct them.

This chapter contains these sections:

- “About Error Messages” discusses error messages and the basic process for finding and resolving them.
- “Error Files” lists and describes the various files generated by the online tools during the conversion to SGML.
- “About SGML” discusses SGML and provides a simple example of an SGML file.
- “Checking the .full File” tells you how to look at the *.err* files and *.full* file for error messages generated during conversion.
- “Finding and Fixing Errors” explains how to find error messages and warnings in the *.sgm* files, how to pinpoint the location of errors in the FrameMaker files, and how to interpret error messages. This section also lists some common *.sgm* file error messages and their causes.
- “Checking for Display Problems” explains how to review the document for dangling cross-references, dangling glossary items, and typical display problems.

¹For books using the *mif2sgml* translator, see Chapter 4, “Finding and Fixing Online Book Errors,” in the -001 version of this book, *IRIS InSight Professional Publisher User’s Guide*.

About Error Messages

While your book is in development, even if your *make* command completes successfully (that is, doesn't terminate prematurely with a *make* or *smake* error), chances are that the content of your book still contains errors. These errors usually involve tags and structures that the translator considers invalid. The conversion tools report such problems as error messages in the SGML output. Before you check in the final version of your book, you must examine the error messages, fix the problems that caused them, and build a clean version of your book with the *make book.full* command. (A "clean" book is one which generates no ERROR messages or link problems during conversion.)

Error Files

Depending on the specific *make* command you run, you can generate a number of different error files. (See "Using make Commands" on page 118 for more information on *make* commands.) The *.full* and *.err* files in this list are the most useful files for error tracking:

.err file An *.err* file contains errors and warnings generated from the corresponding SGML file, as well as information on obsolete or unrecognized tags from the corresponding *.mif* file. You should fix all messages labeled *ERROR*, but you may ignore the messages labeled *WARNING*.

short_title.full file The *make book.full* command generates a file called *short_title.full* that contains a complete error report for the entire book. This is the only error report that contains information on links and unresolved glossary terms.

About SGML

Each *.sgm* file contains the SGML version of its associated FrameMaker file. In SGML, each item is assigned a structural tag, similar to a FrameMaker tag, that describes that item's role in the overall document. So, in SGML, as in FrameMaker, an item (such as a paragraph or a phrase) might be tagged as part of a hanging list, or as a command, or as a chapter title. A word or phrase can have multiple (nested) labels, as happens with a command within a hanging list.

When the conversion tools find something in the FrameMaker file that they don't know how to tag, or when they find an item located where it's not supposed to be (such as a HangingList paragraph tag before the ChapTitle tag), they produce an error message. The example *.sgm* file in "Example SGML File" shows what error messages look like within an SGML file.

If you're internal to SGI, refer to the *Guide to the SGIDOCBK DTD*, 007-0119-001, for more information about SGML.

Example SGML File

This section contains an example FrameMaker document file and the corresponding example *.sgm* file generated by the *make book.full* command.

The FrameMaker File

Figure 7-1 shows an example FrameMaker file. This one-page document contains several paragraph and character tags, including an invalid structure that would generate an error message during online translation. The invalid structure is a set of square-bulleted paragraphs within a set of round-bulleted paragraphs; the `BulletSquareInd` tag should only be used for sub-steps within a step in a numbered list. (See Chapter 3, "Using the IP Templates to Author Documents," for more information on which structures are legal and which aren't in a document.)

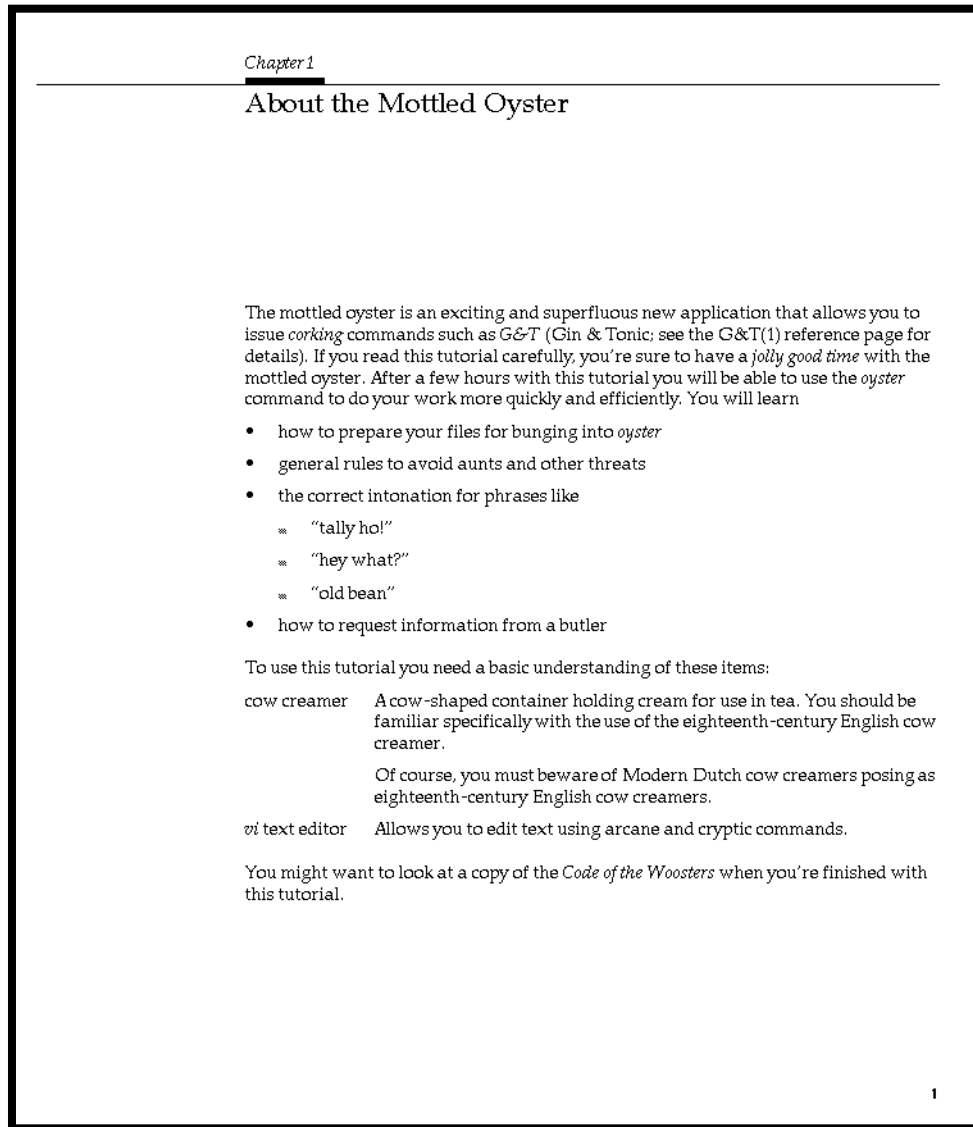


Figure 7-1 FrameMaker Example File

The SGML File

The listing in Example 7-1 represents the *.sgml* file for the FrameMaker file shown in Figure 7-1. Look for names of character and paragraph tags to get an idea of how SGML tags work (“SGML Tag Structure” on page 133 explains the structure of SGML tags in more detail). Notice the *ERROR* lines in the middle of the file. This is how error messages appear within SGML files. The build tools copy all the error messages from each *.sgml* file into a corresponding *.err* file as well as into the *.full* file for your convenience.

Example 7-1 SGML Translation of Example FrameMaker File

```
<!-- Produced by version 4.001 (9/98) of SGI Frame/SGIDOCBK SGML
translator -->
<CHAPTER><TITLE>About the Mottled Oyster</TITLE><PARA>The mottled
oyster is an exciting and superfluous new application that allows you
to issue
<!-- ERROR: (4) Unknown Frame tag "Slang" encountered - detected on
page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "CHAR" TAG
= "Slang" TAGCOUNT = "1" UID = "324065" TEXT = "corking"-->
<DUMMY tag="Slang">corking</DUMMY>
  commands such as <LITERAL>G&T</LITERAL> (Gin & Tonic; see the
G&T(1) reference page for details). If you read this tutorial
carefully, you're sure to have a <GLOSSTERM>jolly good time</GLOSSTERM>
with the mottled oyster. After a few hours with this tutorial you will
be able to use the <LITERAL>oyster</LITERAL> command to do your work
more quickly and efficiently. You will
learn</PARA><ITEMIZEDLIST><LISTITEM><PARA>how to prepare your files for
bunging into <LITERAL>oyster</LITERAL></PARA>
</LISTITEM>
<LISTITEM><PARA>general rules to avoid aunts and other threats</PARA>
</LISTITEM>
<LISTITEM><PARA>the correct intonation for phrases like</PARA>

<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "1" UID = "324242" TEXT = "&ldquo;tally
ho!&rdquo;"-->
<ITEMIZEDLIST><LISTITEM><PARA>&ldquo;tally ho!&rdquo;</PARA>

<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
```

```

<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "2" UID = "324246" TEXT = "&ldquo;hey
what?&rdquo;"-->
</LISTITEM>
<LISTITEM><PARA>&ldquo;hey what?&rdquo;</PARA>

<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "3" UID = "324243" TEXT = "&ldquo;old
bean&rdquo;"-->
</LISTITEM>
<LISTITEM><PARA>&ldquo;old bean&rdquo;</PARA>
</LISTITEM>
</ITEMIZEDLIST>
</LISTITEM>
<LISTITEM><PARA>how to request information from a butler</PARA>
</LISTITEM>
</ITEMIZEDLIST>
<PARA>To use this tutorial you need a basic understanding of these
items:</PARA><DEFLIST><DEFLISTENTRY><TERM>cow creamer</TERM>
<LISTITEM><PARA>A cow-shaped container holding cream for use in tea.
You should be familiar specifically with the use of the
eighteenth-century English cow creamer.</PARA>
<PARA>Of course, you must beware of Modern Dutch cow creamers posing as
eighteenth-century English cow creamers.</PARA>
</LISTITEM>
</DEFLISTENTRY>
<DEFLISTENTRY><TERM><LITERAL>vi</LITERAL> text editor</TERM>
<LISTITEM><PARA>Allows you to edit text using arcane and cryptic
commands.</PARA>
</LISTITEM>
</DEFLISTENTRY>
</DEFLIST>
<PARA>You might want to look at a copy of the <CITETITLE>Code of the
Woosters</CITETITLE> when you're finished with this tutorial.</PARA>
</CHAPTER>

```

The usual method of finding such errors is to look at the *.err* output files after issuing a *make* command. However, you may sometimes need more information about where exactly in a chapter the error occurs. The simplest approach to finding an error in context is to use a text editor (such as *emacs*, *vi*, or *jot*) or a pager (such as *more* or *less*) to locate all occurrences within a file of the word "ERROR".

SGML Tag Structure

You can see from the example in “The SGML File” on page 131 that SGML tags come in pairs. Each tag pair contains an *opening tag* and a *closing tag*, and the opening tag applies to all of the content between it and the closing tag. Figure 7-2 shows an example of SGML opening and closing tags with tagged content, in this case simply the word “oyster.”

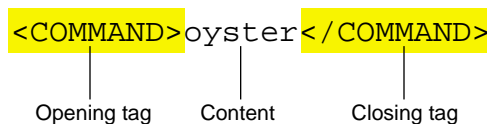


Figure 7-2 SGML Opening and Closing Tags

Checking the .full File

The *short_title.full* file provides a complete list of all the errors, warnings, and link problems in your book. This list consists of four separate kinds of reports:

- the translation error and warning report (one for each FrameMaker file)
- the internal SGI doc report listing the defined parameters from the Makefile
- the link QA report file (a single report for the entire book)
- the glossary QA report file (a single report for the entire book)

Each report starts with three equals signs (“===”) to set it off from the previous report.

You must fix all the errors listed in the error and warning report and the link QA report. You should fix the errors listed in the Glossary QA report, but if you don’t fix them your book will still build.

This section begins with an example *.full* file, then goes on to explain how to deal with each of the three types of report files contained in a *.full* file.

An Example *.full* File

This section contains the *.full* file corresponding to the example FrameMaker file shown in Figure 7-1 (for the purposes of this example, assume that the single page of FrameMaker text is an entire book).

```
=== oyster.err Translation Error/Warning report ===
<!-- ERROR: (4) Unknown Frame tag "Slang" encountered - detected on
page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "CHAR" TAG
= "Slang" TAGCOUNT = "1" UID = "324065" TEXT = "corking"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "1" UID = "324242" TEXT = "&ldquo;tally
ho!&rdquo;"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "2" UID = "324246" TEXT = "&ldquo;hey
what?&rdquo;"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "3" UID = "324243" TEXT = "&ldquo;old
bean&rdquo;"-->

=== Mottled_Oyster Internal SGI Doc report ===

TITLE = Mottled_Oyster
FULL_TITLE = "Mottled Oyster User's Guide"
VERSION = 000-0000-000
  WARNING: VERSION string 000-0000-000 does not match dir oysters
BOOKSHELF = /usr/share/Insight/library/SGI_bookshelves/SGI_Developer
BOOK_ICON = sgi_developer
BOOK_LANG is undefined; default is "C"
DTD = SGIDOCBK
SORT_ORDER is undefined; default is ""
  Note: this is appropriate except for a few special cases
HIDDEN is undefined; default is "FALSE"

=== Mottled_Oyster Link QA report file ===

Link targets appearing multiple times in book: none
```

```

Unresolved References present in this book: none

External Book References present in this book: none

=== Mottled_Oyster Glossary QA report file ===

glossQA: warning - cannot find term (locally): jolly good time

glossQA: checking global glossary
(/hosts/bonnie.engr/depot/doc/1000/007-1859-060/gloss.sgm)

glossQA: 1 glossary references found, 1 undefined terms for
Mottled_Oyster.sgml

glossQA: error - cannot find term: jolly good time

```

The Translation Error and Warning Report

The translation error and warning report contained in the *.full* file provides a complete list of all the translation errors and warnings in your book, grouped by chapter. Before you hand off your final book to be incorporated into a software build, you must fix all the translation errors so that no error messages appear in the translation error and warning report. You do not need to fix warnings (marked as *WARNING* instead of *ERROR*); you can ignore them, but you might want to look at pages where warnings occurred when you review the book in IRIS InSight to catch any display problems).

Here's the translation error and warning report for the example FrameMaker file shown in Figure 7-1.

```

=== oyster.err Translation Error/Warning report ===
<!-- ERROR: (4) Unknown Frame tag "Slang" encountered - detected on
page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "CHAR" TAG
= "Slang" TAGCOUNT = "1" UID = "324065" TEXT = "corking"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "1" UID = "324242" TEXT = "&ldquo;tally
ho!&rdquo;"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->

```

```
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "2" UID = "324246" TEXT = "&ldquo;hey
what?&rdquo;"-->
<!-- ERROR: (77) BulletSquare tags should only be nested within an
ordered list - detected on page 1 -->
<!-- ERRORLOCATION: PAGE = "1" SRC = "oyster.mif" TAGTYPE = "PARA" TAG
= "BulletSquareInd" TAGCOUNT = "3" UID = "324243" TEXT = "&ldquo;old
bean&rdquo;"-->
```

The report includes, for each error, the error message itself (the part that starts with “<!-- ERROR:”) and the chunk of SGML that contains the error (the part that starts with “TEXT =”). The error message briefly describes the problem and gives a page number for the page in the FrameMaker file where the error occurred. You can see which file the error is in because you get a separate error report for each FrameMaker file.

In this example, the problem is that the writer has tried to use the `BulletSquareInd` paragraph tag to create a bulleted list within a bulleted list. This is an illegal structure. To fix this problem, the writer must open the FrameMaker file and re-tag the offending paragraphs with the `DashInd` paragraph tag. Square bullets should only be used to mark sub-steps within a numbered list; `DashInd` should be used for sub-bullets within a round-bulleted list. See Chapter 3, “Using the IP Templates to Author Documents,” for more information on which structures are legal and which aren’t within a document.

The Internal SGI Doc Report

The internal SGI doc report is the second report contained in the *.full* file. It lists the following defined parameters from the Makefile:

- TITLE
- FULL_TITLE
- VERSION
- BOOKSHELF
- BOOK_ICON
- BOOK_LANG
- DTD
- SORT_ORDER
- HIDDEN

For any parameters not defined in the *Makefile*, the default parameter is used and indicated in this report.

The Link QA Report

The link QA report is the third report contained in the *.full* file. It lists multiple link targets, unresolved cross references, and all externally cross-referenced books (it just lists the short title and a 5-digit MIF marker ID for each book; it doesn't verify the links.) Here's the link QA report for the FrameMaker file shown in Figure 7-1:

```
=== Mottled_Oyster Link QA report file ===  
  
Link targets appearing multiple times in book: none  
  
Unresolved References present in this book: None  
  
External Book References present in this book: None
```

In this example, there were no multiple link targets, unresolved cross-references, or externally cross-referenced books.

Multiple Link Targets

Multiple cross references targets usually occur when a writer copies a chunk of text, including the target markers, from one document file and then pastes it into another document file. The example described here occurs when copying figures from one location in a book to another.

Once a figure title (tagged as FigTitle) is created in FrameMaker, a cross reference can be made to its FigTitle. Figure 7-3 shows an example of a figure title that has no cross reference linking to it.

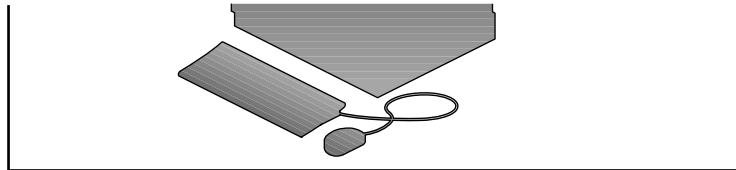


Figure A. An Example Figure Without a Target Marker

Figure 7-3 Figure Without a Cross Reference Target Marker

Figure 7-4 shows a figure title after a cross reference target marker was made. Notice the bold T symbol after the text “Figure A.” The T symbol is the marker that is automatically generated when a cross reference is created.

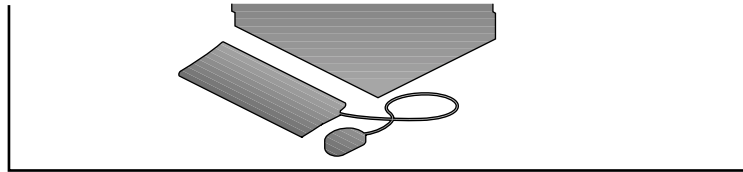


Figure A **T**An Example Figure With a Target Marker

↙ Cross reference (target) marker

Figure 7-4 Figure with a Cross Reference Target Marker

Once an anchored frame and its figure title and the cross reference are created, a writer might copy the figure and figure title (and the cross reference marker) and then paste it somewhere else in the document. If you were to look at the text for the marker in Figure 7-4, it would look like this:

24340: FigTitle: Figure\A Figure with a Cross Reference Target Marker

You can see this for yourself in a Frame document by selecting the marker and choosing Special > Marker. The number 24340 is the unique identifier that is used by the online build tools to create targets. If the marker is copied, the unique identifier is copied with it. So if the figure title for Figure 7-4 was copied and used for Figure B, the marker for Figure B would look the same:

24340: FigTitle: Figure\A Figure with a Cross Reference Target Marker

Notice that the number at the beginning of the string is exactly the same. This poses a problem for the tools, which look at this number and not the title. The tools ignore the figure number and title that show up in the marker box. The tools create a figure number and title from other information supplied, such as autonumbering.

Note: FrameMaker 5 and FrameMaker 5.5.6 do not allow the pasting of a duplicate MIF marker within the same document. If content is copied with the MIF marker and then pasted elsewhere in the same file, the content will be pasted, but the marker will not be.

Finding and Fixing Multiple Link Targets

In the *.full file for the 4.2 build tools, there is a section that identifies multiple cross reference targets. The first step in fixing the problem is to identify the targets.

In the *.full file you may get output that looks like the following:

```
Link targets appearing multiple times in book:
-----
'LE97649-PARENT' :
'LE97649-PARENT' :
'LE97649-TITLE' : LE97649-PARENT How to Use This Tutorial
'LE97649-TITLE' : LE97649-PARENT How to Use This Tutorial
```

Notice that the five-digit number after the “LE” at the beginning of each string is in the list at least twice. The error report lists only the unique identifier and the title, and nothing else.

To locate which FrameMaker document has these multiple target markers, perform the following steps:

1. Use *grep* to determine which document(s) contain the target markers.

```
grep 97649 *.sgm
```

This should produce something that looks like the following:

```
oyster.sgm:<PARA>See <XREF LINKEND="LE97649-TITLE"> for more
information.</PARA><SECTION ID="LE97649-PARENT"><TITLE
ID="LE97649-TITLE">How to Use This Tutorial</TITLE><PARA>To use this
tutorial you need a basic understanding of these items:</PARA>

oyster2.sgm:<PARA>See <XREF LINKEND="LE97649-TITLE"> for more
information.</PARA><SECTION ID="LE97649-PARENT"><TITLE
ID="LE97649-TITLE">How to Set Up an Oyster Farm</TITLE><PARA>First,
you'll need to acquire a stretch of beach near some prime mottled
oyster beds.</PARA>
```

Notice there are four locations where this ID number occurs. The number shows up twice in *oyster.sgm* and twice in *oyster2.sgm*. The XREF LINKEND is the cross reference and the SECTION ID or TITLE ID is the cross reference target. Because there are two targets using the same cross reference ID, this configuration will not work.

2. Either remove the target in *oyster.sgm* or the target in *oyster2.sgm* and establish a new cross reference to the target.

Remove the target that has the fewest cross references pointing to it.

When a cross reference is clicked in the online version of the book, it starts searching the document from the beginning for the matching target and stops at the first one it reaches. For the information listed above, clicking the cross reference for LE97649 jumps to the section “How to Use This Tutorial” because it occurs first in the document.

Table 7-1 Cross Reference Target Scenarios

Scenario	Situation	Solution
A document has two targets and one cross reference	If the cross reference refers to the first target, then the document will work	Delete the target that is not being used
A document has two targets and one cross reference	If the cross reference refers to the second target, then the document will not work	Delete the target that is not being used
The document has two targets and several cross references pointing to one target and only one cross reference pointing to the other target	Varies with the position of the targets within the book.	Delete the target that has only one cross reference pointing to it

Unresolved References

Some unresolved cross references that FrameMaker doesn’t notice are trapped by the build tools. Usually such undetected cross references are caused by remnants of incomplete conversions to a new template or from third-party documentation. You can check for remnant cross references using the Edit > Find/Change tool to search for “Any Cross-Reference.” The link QA report error indicates the approximate location of unresolved references. If you have problems finding the unresolved reference, try using *grep* to find the MIF or *.sgm* files containing the referenced FrameMaker marker ID, then look in those files after the given marker ID. Sometimes there’s an extraneous FrameMaker marker containing a cross reference or cross-reference target, and sometimes the link QA report identifies a nearby marker rather than the marker with the problem.

External Book Reference

The QA report lists the short titles and corresponding 5-digit unique IDs for each external book link included in your document. Review the short titles to ensure they are correct for the books you want to link to from your book.

Glossary QA Report

Unresolved glossary terms are words that are tagged with the “GlossaryItem” character tag but have no entry in either the local or the global glossary. If you’re using the command *make book.full*, you are automatically using the script *glossQA* to validate your glossary terms. *glossQA* defaults to a predefined, internal-to-SGI global glossary. To set up *glossQA* to point to a different location, you can define an environment variable, `GLOBAL_GLOSS`, to override this default. If you have global glossaries for several languages, then you can define the `GLOSS_$LANG` environment variable for each language and use *make book.full* to define the `GLOBAL_GLOSS` environment variable automatically.

Here’s the glossary QA report for the example FrameMaker file shown in Figure 7-1.

```
=== Mottled_Oyster Glossary QA report file ===

glossQA: warning - cannot find term (locally): jolly good time

glossQA: checking global glossary
(/hosts/bonnie.engr/depot/doc/1000/007-1859-060/gloss.sgm)

glossQA: 1 glossary references found, 1 undefined terms for
Mottled_Oyster.sgml

glossQA: error - cannot find term: jolly good time
```

Resolve such errors by creating an entry in the glossary or by removing the GlossaryItem tag from the word in question.

Finding and Fixing Errors

This section discusses error messages that you may encounter while building a book, and explains how to find and fix the corresponding errors in your FrameMaker files.

Types of Errors

There are three general categories of errors that can occur during the bookbuild process:

- *make book* errors (see “make book Error Messages” in Chapter 6 for details)
- MIF-to-SGML translation errors (see the following sections for details)
 - “Critical Input or Internal Problems (1-99)” on page 143
 - “Structural Problems (100-199)” on page 149
 - “Marker and Related Problems (200-299)” on page 153
- unknown errors (see “Unknown Errors” on page 158)

Error messages for *make book* errors appear in the shell window in which you enter the *make* command. Error messages for MIF-to-SGML translation errors and unknown errors appear in files generated during the build process.

Errors are listed in several different files. As explained in “Checking the .full File” on page 133, all the errors from all the files are listed in the *short_title.full* file, so that file is a good place to start looking for errors. However, once you have a general idea of what errors (and how many) are present in your book, you may prefer to work from a file with smaller scope, such as the *.err* file for a particular chapter, and to look in individual *.sgm* files to find the context a given error occurred in. “Error Files” on page 128 describes the different files generated by the conversion tools, and “Using make Commands” on page 118 explains how to use the *make* command to generate a specific type of file.

Finding the Errors in FrameMaker Files

To find a given error, open the FrameMaker file that contains the error, go to the listed page number, and look for the problem. “MIF-to-SGML Translation Errors” on page 143 lists some common numbered error messages and their causes, so look for your error message in that list.

Often the problem is easy to spot: an empty anchored frame or a mis-formatted table, for example. Sometimes, however, the problem is more difficult to find. “Unknown Errors” on page 158 lists some tips to help you find more obscure errors.

MIF-to-SGML Translation Errors

If the translator thinks it knows what the problem is, it provides an error message that looks something like this:

```
ch1.sgm: <!-- ERROR: INFORMATIVE, HELPFUL ERROR MESSAGE HERE -->
```

or

```
ch4.sgm: <!-- WARNING: STOP YOUR SLACKIN' -->
```

This type of error is called a *known error*, because it comes with an informative error message. This section contains a sample of known error messages with explanations and suggestions for fixes. The messages are listed in numerical order, according to the number in parentheses at the beginning of each message. If your error message isn't listed here, follow the instructions in “Unknown Errors” on page 158.

Note that error messages and warning messages generated during MIF-to-SGML translation have the same basic structure; however, error messages usually indicate that the book won't build properly, while you can safely ignore warning messages if you prefer.

Critical Input or Internal Problems (1-99)

MSG NO: 1

Unknown Frame tag [tag_name] encountered

There is an unrecognized paragraph or character tag used in a table or in your document that is not supported by the translator.

Solution: Don't use the tag.

- Search for the unrecognized tag in the document file and replace it with a recognized one.
- If the tag is not found, see if the table uses an unrecognized table format and replace it with a recognized format.

MSG NO: 2

Maximum number of table footnotes exceeded

A table can only have 26 footnotes.

Solution: Use fewer footnotes or split the information into multiple tables.

MSG NO: 3

"Character tag [tag_name] spans Hanging List delimiter"

Character tags shouldn't be applied to whitespace (such as tabs and hard and soft returns). This error message indicates that you've applied a character tag to both parts of a hanging list, including the tab that separates them.

Solution: Apply Default Paragraph Font to the whitespace between the two parts of the hanging list. Tag each part separately.

MSG NO: 4

"Illegal structure starting at '[tag_name]'"

A major structural error occurred.

Solution: Review the location where the error occurred for potential problems.

MSG NO: 5

"XREFNAME has a value [value]"

An internal warning that information saved about a cross reference was not used. After the translator is completed, this message may be disabled.

Solution: If no cross reference errors exist in the *.full* file, ignore this message.

MSG NO: 6

"An ID Value [value] was not used"

An internal warning that should never occur. If it occurs, it is very likely that a cross reference error will exist in the *.full* file.

Solution: Contact techpubs@sgi.com for assistance.

MSG NO: 7

"Use of override is not allowed"

The <override> tag is not used in the SGIDOCBK DTD and cannot be supported.

Solution: Reformatted the affected part of the document to avoid using the override marker.

MSG NO: 8

"Illegal tag in variable: [value]"

A supported tag name could not be found in the listed value. This should be a very rare error.

Solution: Examine the document for potential problems. You can use character tags in variables, but you can't use any character tags that you couldn't use in ordinary text.

MSG NO: 10

"Tag [tag_name] is obsolete or print-only"

The specified tag is no longer supported or else should be tagged as PrintOnly to avoid translation for online display. Use of the tag may work, but should be replaced with an appropriate supported tag when possible.

Solution: Don't use the tag for online content. Find and replace all occurrences of such tags using FrameMaker's Edit > Find/Change tool.

MSG NO: 11

"BNFTerm and BNFRule are obsolete and shouldn't be used"

The BNFTerm and BNFRule tags are no longer included in the IPTemplates used by SGI; however, the SGIDOCBK DTD can still translate them.

Solution: Either replace the obsolete tags with a different paragraph tag such as HangingList or use a TabularList to format the information, or ignore this message.

MSG NO: 20

"INTERNAL: No error message for AV_error"

The AV_error() function was called when the AV_msg string variable was not set.

Solution: This is a programming bug and shouldn't occur for end users.

MSG NO: 21

"Untranslated Function Character ([char])"

The translator was unable to convert a special character value into a character entity.

Solution: Search the FrameMaker document for the special character—it may be easiest to find its location by first using *grep* to search for *character_name* in the MIF file—and replace it with a standard character. If the character must be used, contact the SGI tools people to update the translator or don't use the character.

MSG NO: 22

"INTERNAL: Unclosed Tags [one or more tags]"

This indicates a serious parsing problem occurred and is usually caused by an item being tagged that should not be tagged or by an uncompleted sequence of more than one tag. For example, your file might contain a hanging list that includes a list item without a body.

Solution: Check the tagging and ensure all the components of a structure are present. If the MIF file is valid, then the bug must be fixed in the program.

MSG NO: 23

"Cannot process Frame on page X [tag_value]"

An anchored frame cannot be processed because it was set up incorrectly or because it is empty.

Solution: Verify that the anchored frame is question is set up correctly and import a figure into the anchored frame. See Chapter 5, "Working With Figures," for instructions. If a figure isn't ready, you can ignore this error message, view your book without the figure, and import the figure when it becomes available.

MSG NO: 24

"Cannot process Indirect Frame [tag_value] on page X"

A problem occurred while processing the specified frame, usually a margin figure.

Solution: Double check that the frame is used correctly. Margin figures are no longer supported by the translator, so attach the anchored frame to a Fig tag and set it up as you would an inline figure. See Chapter 5, "Working With Figures," for instructions.

MSG NO: 25

"INTERNAL: MIF Stack Overflow!"

There is a predefined limit to the nesting of MIF tags. In normal usage that limit should never be exceeded.

Solution: Contact techpubs@sgi.com for assistance.

MSG NO: 26

"INTERNAL: MIFPopPgftag on empty stack!"

The input MIF file is syntactically incorrect.

Solution: Use FrameMaker to verify if the MIF file is valid or recreate a new one from the original *.doc* file.

MSG NO: 27

"INTERNAL: Unrecognized MIF Command Name"

An unknown MIF tag type occurred. This is different than an unknown paragraph type occurring.

Solution: Check that the MIF file is valid.

MSG NO: 28

"INTERNAL: MIFPushPgftag stack is full!"

The predefined limit for the nesting of paragraph types was exceeded. This should never occur in normal usage.

Solution: Verify that the MIF file is valid or contact techpubs@sgi.com for assistance.

MSG NO: 29

"INTERNAL: Negative Nesting Level"

This error should never occur with a valid MIF file.

Solution: Check that the MIF file is valid or contact techpubs@sgi.com for assistance.

MSG NO: 30

"INTERNAL: Unmarked data encountered"

The translator found MIF data that it could not process. The data is ignored unless other actions are taken.

Solution: Contact techpubs@sgi.com for assistance.

MSG NO: 31

"Structural loop in setting element [element] [more information]"

The SGML structure programmed into the translator is flawed and causing an endless loop of elements.

Solution: Contact techpubs@sgi.com for assistance.

MSG NO: 32

"Illegal element [element] [more information]"

The SGML structure programmed into the translator is flawed.

Solution: Contact techpubs@sgi.com for assistance.

MSG NO: 33

"Unknown Generic Identifier: [element]"

The translator code specified an unknown SGML element.

Solution: Contact techpubs@sgi.com for assistance.

Structural Problems (100-199)**MSG NO: 100**

"GlossDiv, must occur before any glossary entries"

The GlossDiv structure is designed to be a container of multiple GlossEntry structures. If GlossDiv is used, it must be used before any glossary entries in order to correctly contain them.

Solution: Move the first GlossDiv paragraph before the first glossary entry or remove all GlossDiv paragraphs.

MSG NO: 101

"Glossary tag usage in this context probably incorrect"

The use of a GlossaryDef tag or a GlossaryEntry tag has occurred outside of a glossary.

Solution: Move any GlossaryEntry paragraphs into the glossary document or retag the affected paragraphs with more appropriate paragraph tags.

MSG NO: 110

"Poorly formatted partnumber paragraph. Check for a correct partnumber xxx-xxxx-xxx"

The translator was unable to find all three parts of the part number in the DocNumber paragraph (in the *front.doc*).

Solution: Make sure the paragraph has a correct part number in the form of xxx-xxxx-xxx.

MSG NO: 111

"Structure problem in revision"

The revision paragraph is formatted such that it is impossible to determine what content is the rev number and date (in the *RecRev.doc*).

Solution: Verify that the revision paragraph is formatted correctly.

MSG NO: 112

"The Heading[2,3,4,5] tag occurred without the proper parent headings."

Heading paragraphs must be nested in the proper order. For example, this error could be caused by a Heading3 occurring after a Heading1 without a Heading2 in between.

Solution: Correct the ordering of the HeadingX paragraphs.

MSG NO: 113

"Example or Code captions should come at the beginning of the block"

The title for an Example or Code block must come before the example or code block of text.

Solution: Move the title so it is placed directly before the respective block of example or code text.

MSG NO: 114

"MsgExpl used without preceding Msg tag"

The Msg paragraph tag is part of a construct for displaying error messages that consists of one or more Msg* paragraph tags followed by one or more MsgExpl paragraph tags. This error indicates that the MsgExpl tag was used without a preceding Msg tag. The MsgExpl paragraph can only be used when it follows either a Msg[1st] or MsgMargin[1st] paragraph.

Solution: Make sure the Msg* paragraphs are tagged and ordered correctly.

MSG NO: 115

"Empty figure PARA!"

A figure is located in a paragraph not tagged as Fig.

Solution: Tag the paragraph containing the figure as Fig.

MSG NO: 116

"Unable to determine figure filename"

The translator couldn't find the information about the figure's filename.

Solution: Verify that the figure is imported-by-reference from the *print* dir and that the Fig paragraph is tagged correctly.

MSG NO: 120

"TableTitle should only occur in a table"

A paragraph tagged as TableTitle has been found outside of a table construct. The TableTitle paragraph can only occur directly before a table.

Solution: Make sure the TableTitle paragraph is located right before the table it labels.

MSG NO: 121

"Table figure not located in a table"

A table figure has occurred outside of a table.

Solution: Make sure the figure in question is located and tagged correctly.

MSG NO: 122

"An empty table footnote occurred. Check to make sure it is tagged as TableFootnote"

A footnote inserted in a table must be tagged with the TableFootnote tag rather than the Footnote tag.

Solution: Tag the paragraph as TableFootnote.

MSG NO: 130

"This list should only be nested within another list"

A nested list paragraph, such as any of the *Ind list paragraphs, must reside within another list.

Solution: Make sure the list paragraphs are ordered correctly.

MSG NO: 131

"BulletSquareInd paragraph should only be nested after a List[1st] paragraph"

The BulletSquareInd paragraph tag is only valid when nested in numbered lists. This requirement ensures the formatted SGML will look similar to the FrameMaker document.

Solution: Make sure a List[1st] paragraph occurs directly before the BulletSquareInd paragraph.

MSG NO: 132

"This list is nested incorrectly. Should be at the third level of indentation."

The list paragraph tags should be nested in the correct order. For example, a ListInd4.1st shouldn't occur directly after a List1st because it skips the level made up by the *Ind paragraphs.

Solution: Make sure the order of the list paragraphs is correct or reorder the list.

MSG NO: 133

"The Frame tag ListInd4[.1st] or BulletInd4 (or ProcTitleInd4) should not occur in this context"

The ListInd4, BulletInd4, and ProcTitleInd4 paragraphs can not be used within a MsgExpl or HangingList.

Solution: Move the affected paragraphs or tag them as an acceptable paragraph type.

MSG NO: 134

"The Frame tag ProcTitleInd should only occur nested within a list"

A list paragraph must be used before the indented ProcTitleInd paragraph in order for the tag to be valid.

Solution: Verify that a List paragraph occurs before the ProcTitleInd paragraph or use the ProcTitle paragraph instead.

MSG NO: 135

"ProcSubTitle should only occur nested within a procedure"

A procedure must be started using the ProcTitle paragraph before ProcSubTitle can be used.

Solution: Ensure this is really a nested procedure that was started with a previous ProcTitle paragraph or change this paragraph to use ProcTitle.

MSG NO: 136

"HanglistInd not legal in this context"

The HangingListInd tag cannot structurally reside in this position in the FrameMaker file. For example, you can't legally put a HangingListInd tag inside a BulletInd list.

Solution: Restructure your list so the tag use is legal.

MSG NO: 140

"Hierarchy problem for [tag_name]"

The translator was unable to create an open tag for the paragraph contents or was unable to create an SGML <PARAGRAPH> tag for a figure definition. This means the paragraph tag or figure definition (using the Fig paragraph tag) has been applied somewhere it's not supposed to be, such as before a ChapTitle. This is a very rare error message.

Solution: Verify that the paragraph structure is correct and that only the ChapNum or AppNum tags occur before a ChapTitle.

Marker and Related Problems (200-299)**MSG NO: 200**

"Duplication of Marker #9"

This message means that a single heading or other cross reference destination has multiple #9 cross reference markers attached to it. The SGML only allows one ID per tag. In this case, only the last marker ID value will actually be added to the SGML. This may break some cross references unless fixed.

Solution: Determine which marker is incorrect and remove it.

MSG NO: 203

"No Inline type specified"

This message indicates a problem with your specification of an inline media object. The allowed types are SGIVIDEO, SGIRGB, SGIAUDIO, and SGIINVENTOR. It's also possible that there's a problem with your syntax: an inline media object should be specified as

type:referenced_filename

If you leave out the colon or introduce extraneous spaces or mistype the type or filename, you may get an error message.

Refer to "InSight Inline Object Tags" in Chapter 3 for more information.

Solution: Reformat the paragraph correctly.

MSG NO: 204

"No InlineObj for InlineTitle"

An InlineTitle can only appear after a InlineObj paragraph.

Solution: Make sure a valid InlineObj paragraph exists.

MSG NO: 210

"Illegal tag in index marker: [tag_name]"

An unsupported character tag was specified in an index term. You can use character tags in index markers, but you can't use any character tags that you couldn't use in ordinary text.

Solution: Remove the unsupported character tag.

MSG NO: 211

"Empty index term is ignored."

Part of an index term marker seems to have no text.

Solution: Check the index term marker to ensure it has the correct text between the index term delimiters.

MSG NO: 212

"Only 3 levels of index term nesting is supported"

A single index term can only have three levels of nesting (for example, primary, secondary, and tertiary). This error is caused by having too many ':' delimiters within a single index term.

Solution: Restructure the index term to have, at most, three nesting levels. If a ':' appears in the index term text, and it is not a delimiter, insert a '\' character in front of it.

MSG NO: 213

"Only 3 levels of index term 'sort as' nesting are supported"

This is a very similar to error #212.

Solution: Remove the extra 'sort as' information from the index term.

MSG NO: 214

"Footnote INDEX targets (avoid)"

Index term markers cannot be used within a footnote.

Solution: Remove the index marker from the footnote and move it to another relevant part of the document.

MSG NO: 215

"No matching ']' for '[' in [term] The index term will be ignored"

The square brackets within an index term are unmatched. This error makes it impossible to correctly parse the index term.

Solution: Correct the syntax of the index term. Refer to "Cross References" in Chapter 3 and the FrameMaker documentation for more information on creating index markers.

MSG NO: 216

"Fatal index term error in [term] The index term will be ignored"

The index term does not have the correct syntax for an index marker.

Solution: Correct the syntax of the index term. Refer to "Cross References" in Chapter 3 and the FrameMaker documentation for more information on creating index markers.

MSG NO: 220

"Invalid Marker 18 format (EXTREF)"

The content of marker #18 for external book links does not have a space character that delimits the short-title and the ID number. The correct syntax is:

`<target_book_short_title><space><target_element_unique_id>`

Solution: Correct the marker #18 content.

MSG NO: 221

"Marker 19 occurred without a preceding Marker 18"

A marker type 18 with the correct content to start an external book link must occur before the marker type 19 that is placed at the end of the link text. The proper syntax is:

`<marker#18>text<marker#19>`

Solution: Make sure the proper markers are in the correct locations and that the document title and the markers are tagged as DocumentTitle.

MSG NO: 222

"Marker 17 data required for LaunchWord"

A marker type 17 must have content to create the launchable link. The preferred syntax is:

`Launchword:app:parms`

Solution: Correct the marker #17 content. Refer to "Creating Links to URLs and Applications" on page 90 for more information.

MSG NO: 230

"Illegal page number cross reference"

A cross reference that shows only the generated text of a page number is useless within an online display environment. The link is broken and should be replaced with a more appropriate cross reference type.

Solution: Use a different cross reference format.

MSG NO: 231

"Unknown cross reference type: [cross_ref_type]"

A currently unsupported cross reference type was encountered by the translator.

Solution: Change the cross reference to a more recent type or contact techpubs@sgi.com to see if support for the cross reference can be added.

MSG NO: 232

"Footnote XREF targets (avoid)"

You can't make a footnote the target of a cross-reference.

Solution: Move the cross reference to point to content outside of the footnote.

MSG NO: 233

"Possible unused Frame Target [marker_content]"

The indicated FrameMaker target probably doesn't have anything pointing to it, or a marker #9 was encountered that doesn't have the five unique ID digits at the beginning of the string. It will be impossible to create a cross reference to that destination.

Solution: Remove the unused marker and double-check that the cross reference markers are set up correctly.

MSG NO: 234

"Illegal EXTERNAL REFERENCE file name"

A problem occurred when creating an external book link.

Solution: Ensure the external cross-reference link has been established correctly or use the preferred linking method with Markers 18 and 19.

Unknown Errors

If the translator gets completely confused, it prints an error message that looks something like this:

```
ch2.sgm: <UNKNOWN.ERROR>
```

This type of error is called an *unknown error* and seldom provides a helpful error message. The report lists the page number (for the FrameMaker document) on which the error occurred.

The translator generates an “unknown error” message when the file conforms to the DTD (and is therefore structurally legal) but a problem exists anyway. Several different (but rare) conditions can cause this error. The problem is usually located in the object immediately preceding the `<UNKNOWN.ERROR>` tag; examine the *.sgm* file to see exactly where the `<UNKNOWN.ERROR>` message occurred.

To fix an unknown error, open the FrameMaker file, go to the page number indicated in the error report, and look for anything that you think might be causing a problem. Here are some things to look for (in order of likelihood):

- Review the list in “Tagging FrameMaker Files” on page 73. Make sure you haven’t violated any of these rules.
- Examine any unusual order of elements. For example, if you put a Figure directly after an Example—or a TextInd, Example, or Code directly after a Note—you’ll get an error message.
- Read and follow the guidelines in “Tagging FrameMaker Files” on page 73. Problems often arise from errors in figures and in tables.

Checking for Display Problems

After your book builds with no errors, bring it up in InSight as described in “Viewing a Built Book” in Chapter 6 and scroll through it carefully. Here are a few things to review:

- Do the links work? Do they take you to the right places? If you find a broken cross-reference, check to make sure it’s correctly formatted in the FrameMaker file (see “Creating Online Links” on page 77 for instructions).
- Are the Table of Contents, List of Figures, List of Tables, and Index correctly formatted? If not, try deleting those files and replacing them with the appropriate template files from the InSight Professional Publisher template.
- Do all figures appear in the online text? If not, make sure that you set them up correctly (according to the instructions in Chapter 5, “Working With Figures”). If figures do appear, do they display correctly?

Creating Online Help

This chapter explains how to use FrameMaker and the online bookbuilding process to create online help for an application. The same source file is used for online books and SGI Help; when you're writing a book, you can mark sections of it to be used by the help system.

The following sections are included in this chapter:

- "Overview of Help"
- "Tagging FrameMaker Files for Help"
- "The Helpmap File"
- "Creating A Rollover Menu"
- "Creating Context Sensitive Help"
- "Using the World Wide Web from Your Help Menu"
- "Testing the Online Help"
- "Telling Users Which Subsystems to Install"
- "Troubleshooting"

Overview of Help

The SGI Help system relies upon several main components:

- A FrameMaker file containing special labels that mark the sections you want to use as help topics. When you build the online book, these labels become a part of the SGML file.
- A helpmap file that provides a mapping between the help topics in the online book and the Help menu in the application.
- A help server that handles communication between the application and the helpmap file.

Figure 8-1 shows the flow of how an application accesses its help information.

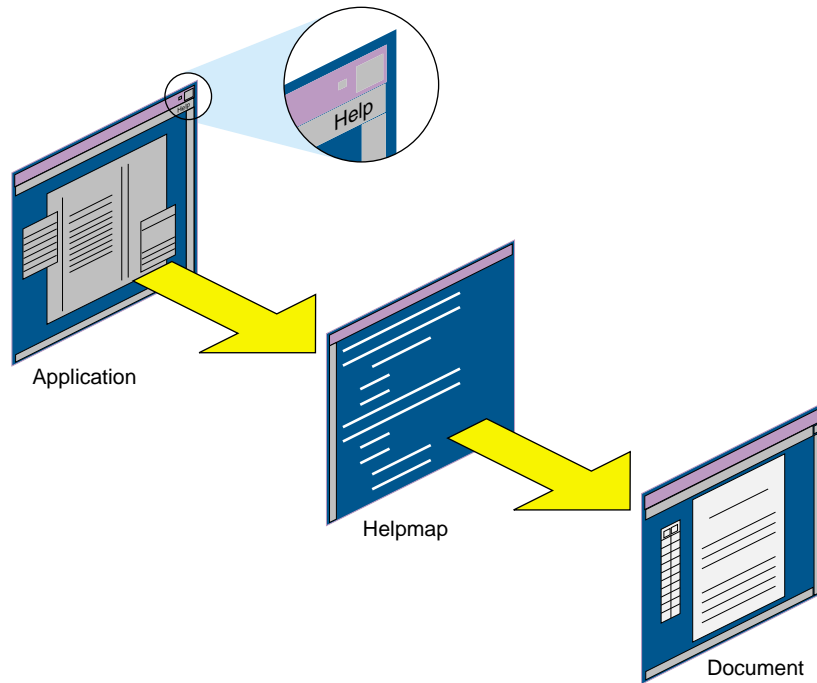


Figure 8-1 Standard Flow for Help

Application

Depending on its complexity, an application can include a help menu as well as a help button (context-sensitive). Widget strings are used to tie the application to its helpmap.

Helpmap

The *helpmap* is the key to making help work correctly. This ASCII text file provides a mapping between the help menu in the application and the help topics in the online book. Each helpmap line contains six fields that define the mapping. The third field of the helpmap corresponds to one of the application's Help pulldown menu items and to a help topic in the FrameMaker source used to create the online document.

Document

The document is the IRIS InSight document being accessed by the helpmap. Any full section of an online document—from one heading to the next heading—can be used as a help topic. Part of a section of a document, as long as it is a complete SGML structural unit, can also be used as a help topic.

Tagging FrameMaker Files for Help

Any full section of a book—from one heading to the next heading of the same or higher level—can be used as a help topic. To mark it as such, place an empty paragraph above the chosen section's heading (any paragraph tagged Heading1, Heading2, or Heading3); then, in this new paragraph, enter a unique one-word label to identify the section. Tag the label with the HelpTopic paragraph tag; then apply the OnlineOnly conditional text tag to it. Here's a more detailed set of instructions:

1. Place the cursor on a blank line above the heading of the section you want to use as a help topic.

If you place it above a Heading1, the help topic will include any Heading2 and Heading3 subsections that the marked section contains. If you place it above a Heading2, the help topic will include any Heading3 subsections that the marked section contains.

2. Type a unique label for that topic. Later, you'll be entering this label into the helpmap file, thereby tying the help topic to the application.

Do not put spaces or hyphens in the label. Ensure that you choose a name that another writer isn't likely to use. For example, you might want to include the application name, or an abbreviation of the name, followed by some other text.

3. Tag the paragraph with the HelpTopic paragraph tag.
4. Choose Special > Conditional Text.
5. Select the entire paragraph containing the new label; then choose OnlineOnly from the Conditional Text dialog. Click the left-pointing arrow button in the dialog to move "OnlineOnly" into the "In:" column; then click the Apply button. Close the Conditional Text dialog.

Parts of sections of a book can also be used as help topics. Note that such pieces must be complete SGML structural units, such as paragraphs or tables or figures; you can't use part of a paragraph, or part of a table, or part of a figure, as a help topic. In fact, if you use a table you must also use the entire paragraph that the table anchor appears in.

To create a help topic from a piece of a section, follow these steps:

1. Place the cursor on a blank line above the first paragraph of the subsection you want to use as a help topic.
2. Type a unique label for that topic, then a colon, then a title for the topic. The label follows the same rules as for HelpTopic, above; the title is necessary since there's no section title to use for the help card title. The label and title are eventually put into the helpmap file by the bookbuilding tools, thereby tying the help topic to the application.

Remember not to put spaces or hyphens in the label, and to choose a name that another writer isn't likely to use.

3. Tag the paragraph with the HelpSubTopic paragraph tag.
4. Choose Special > Conditional Text.
5. Select the entire paragraph containing the new label; then choose OnlineOnly from the Conditional Text dialog. Click the left-pointing arrow button in the dialog to move "OnlineOnly" into the "In:" column; then click the Apply button. Leave the Conditional Text dialog open.
6. Now select all of the text that you wish to appear in the help topic. Choose HelpSubTopic from the Conditional Text dialog. Click the left-pointing arrow button in the dialog to move "HelpSubTopic" into the "In:" column; then click the Apply button. Close the Conditional Text dialog.

Help Buttons Versus Help Menus and Context Sensitive Help

The number of help labels you need to include depends on the complexity of the application and the method by which people will access help. There are two common approaches to online help: a *Help* button or a Help menu.

- If the application has a *Help* button, add a label to your book above the heading of the section that you want to appear when the user clicks the *Help* button. If the application allows you to open several windows, each with its own *Help* button and help card, put a different label above each topic that should come up in response to a *Help* button. Provide the help labels to the application developer so they can be incorporated into the application's code. The application passes this info to the help server when the user clicks a *Help* button.
- If the application has a Help menu, add a label to your book above each section that should appear in the Help menu or should appear when a user requests context sensitive help for a particular area of the window. If the application is based on the IRIS ViewKit interface toolkit, you do not have to pass this information on to the developer. If it is not ViewKit-based, you need to provide the developer with a list of the topics for the menu and all of the associated labels. The application developer will then incorporate these labels into the application's code.

Note: Tag your FrameMaker files the same way regardless of whether you're creating context-sensitive help or menu-based help. The difference between the two is indicated in the helpmap file.

The Helpmap File

The helpmap file acts as the glue that binds the help topics to the application. In this ASCII text file, you list the topics that you've designated as "help" in your FrameMaker files, and you specify when the topic should appear—when a user chooses a particular item from the Help menu, for example, or when the user clicks in a certain place in the window.

The name and location of the helpmap file are important. If the application is based on IRIS ViewKit, name the file *AppName.helpmap*. The AppName should be the same as the name of the application's defaults file in */usr/lib/X11/app-defaults* (usually a capitalized form of the name of the executable). If the application isn't ViewKit-based, the writer and the developer need to agree on a name to use for the helpmap file.

Place the file in your working directory, in a subdirectory called *help*. For example, suppose your book directory is */usr/people/debbie/books/Desktop_UG*. The helpmap file should be placed in */usr/people/debbie/books/Desktop_UG/help*.

An Explanation of the Helpmap File

Each line of the helpmap file contains six fields separated by semicolons. All fields must be in the same line of text (so don't press Enter when you reach the end of the line, even if the line wraps in your ASCII test editor). Here's the format to use, followed by explanations of the fields:

topic_num ; short_title ; title_str ; level_id ; help_key ; widget_string

topic_num	A number that indicates what type of help topic it is: 0: Not on a help menu—either context sensitive or tied to a Help button 1: Overview topic on the Help menu 2: Task-oriented entry that shows up on the Help menu 3: “Keys & Shortcuts” entry on the Help menu
book_name	The short title of the book that contains the help topic.
title_str	The name of the topic. This determines what appears on the Help menu and in the index. This name can be different from the HeadingX title that appears in the FrameMaker file. Be sure to fill in this field even if the topic is one that doesn't appear on a Help menu.
level_id	A number indicating whether the item is a top-level menu entry or on a rollover menu. Use 0 if the item is a top-level entry, or isn't on a menu (as with button or context-sensitive help).
help_key	The label in your FrameMaker file that is associated with this help topic.
widget_string	Specific information that identifies the application, or portion of the application, to which the help topic belongs. For the “Overview” menu entry, use AppName.windowName.overview If the topic is tied to a help button, use AppName.windowname For a list of “Keys & Shortcuts,” use AppName.windowName.keys If the topic is tied to a portion of the window (as is true for context sensitive help), you need the fully qualified widget string. You can include several different widget strings if you want to use the same topic for different regions of the window. See “Creating Context Sensitive Help” and “Setting Up the Fallback Mechanism for Context Sensitive Help.”

A Sample Helpmap File

This section contains a sample helpmap file.

```
1;Desktop_UG;The Desks Overview Window;0;ov_deskoverview;Overview.DesksOverview.overview
2;Desktop_UG;Creating a Desk;0;ov_createdesk;Overview.DesksOverview
2;Desktop_UG;Switching Between Desks;0;ov_switchdesk;Overview.DesksOverview
2;Desktop_UG;Moving Windows Between Desks;0;ov_moveitems;Overview.DesksOverview
2;Desktop_UG;Copying Windows Between Desks;0;ov_copyitem;Overview.DesksOverview
2;Desktop_UG;Listing All the Windows;0;ov_listall;Overview.DesksOverview
2;Desktop_UG;Renaming a Desk;0;ov_renamedesk;Overview.DesksOverview
2;Desktop_UG;Deleting a Desk;0;ov_deletedesk;Overview.DesksOverview
2;Desktop_UG;Placing a Window in All Desks;0;ov_globaldesk;Overview.DesksOverview
2;Desktop_UG;Manipulating Windows;0;ov_manipulatewin;Overview.DesksOverview

2;Desktop_UG;Changing the Overview Display;0;ov_changeviews;Overview.DesksOverview
2;Desktop_UG;Changing the Order of Desks;1;ov2_arrangedesks;Overview.DesksOverview
2;Desktop_UG;Displaying the Names of Items in a Desk;1;ov2_displaynames;Overview.DesksOvervie
w
2;Desktop_UG;Hints for Resizing;1;ov2_resizinghints;Overview.DesksOverview
2;Desktop_UG;Viewing Desks as Snapshots or Buttons;1;ov2_displaybutton;Overview.DesksOverview
2;Desktop_UG;Resizing Snapshots or Buttons;1;ov2_resizebutton;Overview.DesksOverview

0;Desktop_UG;The Desk Display Area;0;ovcon_display;Overview.DesksOverview.MainView.Frame.view
port.Bboard
0;Desktop_UG;The Global Desk;0;ovcon_global;Overview.DesksOverview.MainView.globalDesk
0;Desktop_UG>About Desks Overview Menus;0;ov_aboutmenus;Overview.DesksOverview.menuBar
0;Desktop_UG;The Overview Menu;0;ovcon_overmenu;Overview.DesksOverview.menuBar.Overview
0;Desktop_UG;The Desks Menu;0;ovcon_deskmenu;Overview.DesksOverview.menuBar.Desks
0;Desktop_UG;The Window Menu;0;ovcon_winmenu;Overview.DesksOverview.menuBar.Windows
0;Desktop_UG;The List All Window;0;ovcon_listwindow;Overview.WindowList_popup
3;Desktop_UG;Keys and Shortcuts;0;ov_keyshortcuts;Overview.DesksOverview.keys
```

Creating A Rollover Menu

You can create a rollover menu for a help topic. For example, suppose a Help menu includes the topic “Changing the Overview Display” and you want to include information on various ways in which you can change the display. You set this up through the helpmap file by using a different *level_id* for the rollover menu items. Note the *level_id* of 0 in the first entry and the *level_id* of 1 in subsequent entries:

```
2;Desktop_UG;Changing the Overview Display;0;ov_changeviews;Overview.DesksOverview
2;Desktop_UG;Changing the Order of Desks;1;ov2_arrangedesks;Overview.DesksOverview
2;Desktop_UG;Displaying the Names of Items;1;ov2_displaynames;Overview.DesksOverview
2;Desktop_UG;Hints for Resizing;1;ov2_resizinghints;Overview.DesksOverview
2;Desktop_UG;Viewing Desks as Snapshots or Buttons;1;ov2_displaybutton;Overview.DesksOverview
2;Desktop_UG;Resizing Snapshots or Buttons;1;ov2_resizebutton;Overview.DesksOverview
```

Creating Context Sensitive Help

If an application has a Help menu, you can create context sensitive help. Context sensitive help allows you to provide information about specific areas of controls on a window. For example, the InPerson desktop conferencing software has an area into which you can drop icons that represent the people you want to call. This is called a drop pocket. Context sensitive help might explain what the drop pocket is and how to use it.

To create context sensitive help, you must identify the area of the window to which you want to link the help topic. You do so by giving the full name of the widget in that area of the window. (A widget, in X parlance, is an element of a user interface—a button, scroll bar, text area, or other interface item.) The widget’s full name includes the name of each widget in the hierarchy that contains it, from the entire window down to the specific area in question. For example, here is the widget name for the drop pocket on the InPerson window:

```
InPerson.callWindow.Form.GroupPanel.scroll.iconForm.Xdraw
```

If the application is ViewKit-based, and has been linked to the help server, you can get the widget name relatively easily:

1. Open a shell window.
2. Kill the help server and restart it in debugging mode by typing:


```
% /etc/killall sgihelp
% sgihelp -debug
```
3. Choose “Click for Help” from the Help menu, then click on the region for which you want to provide context sensitive help. The widget string appears in the shell window. Below is an example of what you might see in the shell window:

```
% sgihelp -debug
REQUEST =client="InPerson" command="view" book="" keyvalue="callWindow.Form.GroupPanel.scroll
.iconForm.Xdraw" separator="." user_data="" title="InPerson" subsys="InPerson.books.InPerson_
AG and InPerson.books.InPerson_UG"
INFO FOR HELPMAP =InPerson.callWindow.Form.GroupPanel.scroll.iconForm.Xdraw
```

The INFO FOR HELPMAP field contains the widget name to use in your helpmap.

Setting Up the Fallback Mechanism for Context Sensitive Help

You might not provide context sensitive help for every button or set of controls on a window. However, you need to make sure that something reasonable appears, no matter where a user clicks. For example, if a user clicks on an area for which there isn’t any specific context sensitive help, you might display the Overview help card.

Here’s how it works. When a user requests context sensitive help, the help server looks to see if that widget name appears in the helpmap file. If not, it drops the last item in the widget name and tries again to find a match. It continues doing this until it finds a match. For example, suppose a user is reading a book in IRIS InSight and requests context sensitive help on the *Go Back* button. The widget string that gets reported is:

```
Insight.InsightBook.IvForm.ivPane2.commandArea.DocCmdAreaRC.goBack
```

Suppose help is not available for the *Go Back* button. The help server continues searching and finds a match with this portion of the widget string:

```
Insight.InsightBook.InsightBook.IvForm.ivPane2
```

Here's a helpmap entry which causes the Overview card to display under those circumstances:

```
1;InsightHelp;Overview;0;inshelp_overview;Insight.overview;Insight.InsightBook.InsightBook.IvForm.ivPane2
```

Using the World Wide Web from Your Help Menu

You can include a link to one or more World Wide Web pages in your help menu if you so desire. This feature can be useful, for instance, if you want to provide release notes, frequently asked questions lists, and other such time-sensitive information by way of the Web.

Including a URL in a help menu is much like placing any other item in that menu; it merely requires a line in the helpmap. (Note, though, that not all customers have access to the Web, so you shouldn't put any critical information on a Web page.) The format is identical to other lines in the helpmap file. For *topic_num* and *level_id*, use 0 even though the item will appear on the Help menu. For *short_title*, give the complete URL. For *title_str*, give the entry you want to appear in the Help menu; you should also indicate somehow to the user that this item launches a Web browser. (You can, for instance, use the word "(Web)" in parentheses at the end of the *title_str*.) Leave the *help_key* field blank; there is no help topic in the book that this item corresponds to. And finally, the *widget_string* should be the name of the application's top-level widget. For example:

```
0;HREF=http://dyne.yoyo.com/dougom/Tutorial.html;examples (Web);0; ;Cvpav.cvpav
```

That's all there is to it; when the user chooses this item from the Help menu, a Web browser is launched and the URL is opened.

Testing the Online Help

This section outlines the process of building and testing online help. Once you've checked in the files and the application has been built, you can install the product and test the help. This section explains how to test the help before the product is rebuilt. It does not provide details on the *make book* command or the setup that's required to build an online book. See Chapter 6, "Building Online Books," for that information.

Testing Help From the Application

To test an online book when you have access to the application:

1. Build the book:

```
% make book
```

2. Change directories into the subdirectory where the book has been created:

```
% cd books
```

You should be in a directory named something like
/usr/people/debbie/books/Desktop_UG/books

3. List the contents of the directory:

```
% ls
```

Find the directory named after the book. For instance, in this example, you should see a directory named *Desktop_UG*. This is the directory that contains the online book and help.

4. Become superuser (root):

```
% su
```

5. Copy this online book directory (*Desktop_UG* in the example) into */usr/share/Insight/library/companyname_bookshelves/companyname/books*, where *companyname* is the name or abbreviation for your company (as determined in "Editing the Makefile" on page 110).

6. Edit the associated *booklist.txt* file. For example, if you copied the book files into the directory */usr/share/Insight/library/Yoyo_bookshelves/Yoyo/books*, edit the *booklist.txt* file in */usr/share/Insight/library/Yoyo_bookshelves/Yoyo*.

Make sure that this file includes an entry for the book whose files you copied. If it doesn't, copy the information from the *booklist.txt* file in your working directory.

7. Copy the helpmap file(s) into the directory */usr/share/help*.

8. If you're already running the help system, kill it by typing:

```
% /etc/killall sgihelp
```

It will automatically restart when you next request help.

9. If you're running the application whose online help you want to test, quit out of the application and re-launch it.

10. If the application has a Help menu, select the “Overview” entry to launch SGI Help. See Figure 8-2. If the application has a *Help* button, click it to launch SGI Help.

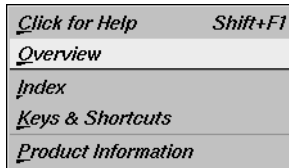


Figure 8-2 A Sample Help Menu

11. Study the information in right viewer pane to ensure that the appropriate help card information for first helpkey is showing. (The first helpkey is usually associated with an overview of the application.) Figure 8-3 is an example of the help for the Apanel application.



Figure 8-3 A Sample SGI Help Graphical User Interface

12. Choose the next item from the Help menu and ensure that the correct help card appears in the right pane.
13. Continue choosing each item in the Help menu and ensure that the correct help cards appear in the right pane.
14. Test the context sensitive help by choosing Help > Click for Help and position and click the question-mark cursor over a particular location of the application’s GUI (scrollbar, menu bar, button, etc.) that you know to have associated help information.
15. Repeat step 14 until you have tested all the context sensitive help.

16. If your system “beeps” and a different help card doesn’t appear in the right viewer pane or if a dialog box appears, then either
 - the corresponding line of the helpmap needs to be fixed.
 - the help topic is missing from the online book (and is probably missing or mis-tagged in the FrameMaker file).

See “Troubleshooting” on page 174 for more information.

Testing Help Without the Application

To test an online help book when the application is not available, follow the directions in steps 1 through 8 in “Testing Help From the Application” on page 171. Then perform the following steps to extract and test the helpkeys directly:

1. In a shell, type:

```
% awk -F\; ' $2 == "short_title" { print $5;}'
< /usr/share/help/App.name.helpmap > ~/App.name.keys
```

This script extracts the information from the fifth field of the helpmap (for those lines where the second field has this book’s *short_title*) and creates a file containing a listing of these helpkeys (these are the same names as *HelpTopic*-tagged paragraphs in a book). Since a helpmap may contain helpkeys for more than just one book, this script extracts helpkeys associated with a particular book. For applications with more than one book as their source of help, run the script again for the other book(s) and give the resulting file a slightly different name.

2. Open and display the text file *App.name.keys*. Reduce the size of the shell and move it to one side of your desktop.
3. Enter the following test script into another shell using the first helpkey from the list. Append the script with *-f &*:

```
% sgihelp -b short_title -k helpkey_name -f &
```

An SGI Help—*App.name* viewer should appear.

4. Study the information in right viewer pane to ensure that the appropriate help card information for first helpkey is showing. (The first helpkey is usually associated with an overview of the application.)
5. Double-click on the next item in the left viewer pane (the Table of Contents) and ensure that the correct help card appears in the right pane.

6. Continue double-clicking on each item in the list in the left viewer pane and ensure that the correct help cards appear in the right pane.
7. If your system “beeps” and a different help card doesn’t appear in the right viewer pane or if a dialog box appears, then either
 - the corresponding line of the helpmap needs to be fixed.
 - the help topic is missing from the online book (and is probably missing or mis-tagged in the FrameMaker file).
8. If your application’s help comes from other books, open and display the other *App.name.keys* text files and repeat steps 3 through 6.

Telling Users Which Subsystems to Install

If a user hasn’t installed the correct online book, an error message will be seen when online help is accessed. To make this error message as useful as possible, you need to ask the application engineers to include two extra lines in the application’s defaults file. Here’s an example of what appears in the *app-defaults* file for the Desks Overview tool:

```
*helpSubSys: desktop_eoe.books.Desktop_UG
*helpTitle: Desks Overview
```

If a user doesn’t have the book installed, a message comes up saying to install the *desktop_eoe.books.Desktop_UG* subsystem.

Troubleshooting

When you begin testing the online help, you may see one of several problems:

- The help server says that it’s unable to find help on a particular topic. See “No Help Available.”
- The wrong section appears when you choose a help topic. See “The Wrong Help Topic Appears.”

No Help Available

If you get a message saying something like “Sorry, no help available on that topic”:

- Check to make sure the book is installed. Look in all the relevant subdirectories of your company’s bookshelves directory under `/usr/share/Insight/library`, including any Help subdirectories.
- Check the generated file `booklist.txt` in the relevant subdirectory of your company’s bookshelves directory under `/usr/share/Insight/library`. Make sure it has an entry for the book, and that the book’s title is spelled correctly.
- Make sure the helpmap file is in `/usr/share/help`.

The Wrong Help Topic Appears

If the wrong section of the book appears when you click the *Help* button or choose a topic, you probably have a problem in the FrameMaker file, or a mismatch between the application, the FrameMaker file, and the helpmap file.

In the FrameMaker file:

- Make sure that the paragraph with the help label is tagged `HelpTopic`.
- Make sure that the heading below the `HelpTopic` is not accidentally tagged with the `HelpTopic` paragraph tag.
- Make sure the text in the `HelpTopic` paragraph is `OnlineOnly` conditional text.

To check for a mismatch between the FrameMaker file, the helpmap file, and the application:

1. Stop the help server and restart it by typing the following:

```
% /etc/killall sgihelp
% sgihelp -debug
```

2. Click on a *Help* button or choose a topic from a Help menu. A bunch of text appears in the shell window. Here are two examples:

```
REQUEST =client="Overview" command="view" book="Desktop_UG" keyvalue="ov_createdesk"
separator="." user_data="" title="Desks Overview" subsystem="desktop_eoe.books.Desktop_UG"
REQUEST =client="Overview" command="view" book="" keyvalue="DesksOverview.menuBar.Desks"
separator="." user_data="" title="Desks Overview" subsystem="desktop_eoe.books.Desktop_UG"
INFO FOR HELPMAP =Overview.DesksOverview.menuBar.Desks
```

client	The name of the application. Make sure the name of the helpmap file and the last entry in the file match this.
book	The short title of the book that contains the help content. Make sure the spelling in the helpmap file and <i>booklist.txt</i> file match this. This field will be empty if you have chosen “Click for Help.”
keyvalue	One of two pieces of information: either the HelpTopic tag in your FrameMaker file and helpmap file or a portion of the widget string for the context sensitive help. If it’s the label or tag you’ve used to mark the help topic, check the helpmap file to make sure the spelling, capitalization, and punctuation match.
INFO FOR HELPMAP	The widget string to be used for context sensitive help. Make sure this matches the widget string you’ve placed in the helpmap file.

Missing Items on the Help Menu

Check the helpmap file if items are missing from the Help menu. Check the following:

- Make sure the line in the helpmap file begins with the number 2. This specifies that it is a topic that should appear on the menu.
- Make sure you haven’t left out a semicolon between any of the field.

IRIS Insight Bookbuilding Architecture

This appendix describes the current IRIS InSight book-building architecture and process.

Bookbuilding Architecture

Figure A-1 shows the general process flow for building the textual content of an IRIS InSight book. Following the figure is a more detailed text description of the process.

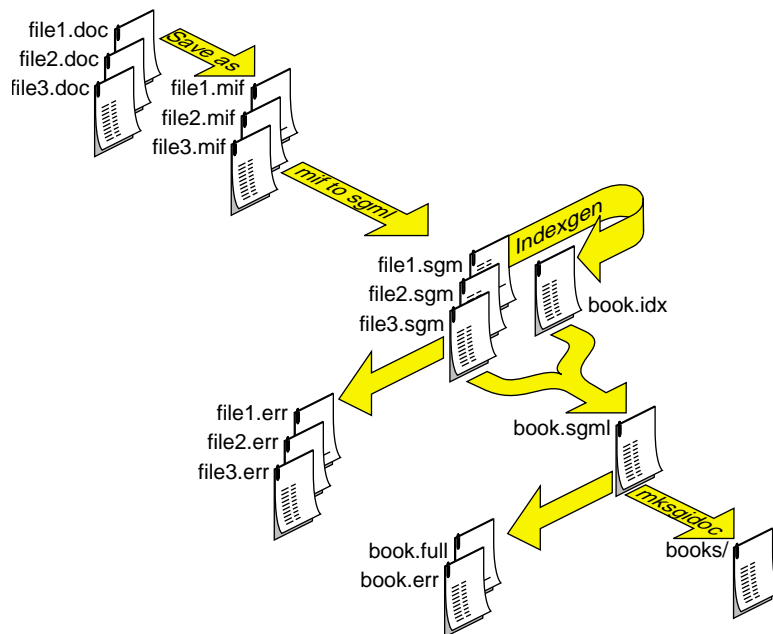


Figure A-1 Architecture for Book-Building

Authors compose a book using FrameMaker, then “compile” the book into an InSight viewable book:

- The FrameMaker files (*.doc*) are converted to Maker Interchange Format (*.mif*) using FrameMaker’s *fmbatch* utility. A special file, *conditional.doc*, is used during this process to force the proper FrameMaker conditionals to be applied.
- The next phase is to translate the MIF files (*.mif*) into SGML files (*.sgm*) using a MIF to SGML translator such as *mif2sgml*, a translator from SGI. After each file is translated, the *.sgm* file is scanned to see whether it contains any CGM figure references. CGM figures are generated from FrameMaker line art. If the *.sgm* file contains CGM figure references then it is run through Inso’s *miftoctgm* utility and every anchored frame is converted to a CGM image. The file is scanned again to convert the CGM images to GIF images for use by IRIS InSight.
- Once all the SGML files have been generated for the book, they’re concatenated and the *indexgen* utility is run over them to create a back-of-the-book index. This index is then concatenated to the rest of the SGML files, and a document bounding tag pair, such as `<SGIDOCBK> </SGIDOCBK>`, is wrapped around the result to create the master SGML file that is fed into Inso’s *mksgidoc* or *mksgidocbk* command to create an IRIS InSight viewable book.

Note: Supported DTDs include *SGIDOCBK* and *SGIDOC*. Arbitrary DTDs are supported to a limited degree; for example, *CDOCBK* will work with the *mkbook* command, but the *indexgen* utility is not supported.

Glossary

blue italic

The color and style of an online glossary term. Click on the word to launch a restricted viewer containing its definition.

bookbuild

The process of converting a book from FrameMaker files to *SGML* files using IRIS InSight Professional Publisher tools. Sometimes called a “build.”

CGM

Computer Graphics Metafile image file format. An “infinite-resolution” image format, relying on shape descriptions rather than bitmaps.

chapter tabs

Pages used in a hard-copy book to mark the beginnings of chapters. Tabs often have shaded edges so that a reader can easily find them by looking at the edge of the book.

chunks

Small units of text, written so as to make sense without surrounding context. Text intended for online viewing should be chunked as much as possible, since there’s no guarantee that a reader will have read any surrounding sections.

closing tag

The second tag in a pair of *SGML* tags. The corresponding *opening tag* stops applying to subsequent text after the closing tag.

composite figures

Figures in a FrameMaker file which consist of more than one image—either multiple imported image files in your book directory, or a combination of FrameMaker line art and one or more imported image files. Note that it’s often best to create a composite image outside of FrameMaker—using IRIS ShowCase, for instance—and snap the result, in order to import only one file into a given frame in FrameMaker and thereby make color separations easier. However, it’s legal to import multiple files into a single frame should you wish to do so.

DTD

Document Type Definition: a formal description of the allowable structure of a document. Specifies, for instance, what tags can follow what other tags.

entity

In *SGML* jargon, an entity is a named definition of a character, usually named with a cryptic-looking string like ““” (which indicates a left double quotation mark).

known error

An error during the *bookbuild* process with an obvious cause, recognized by the build tools.

MIF

Maker Interchange Format: an ASCII format for FrameMaker files which makes it easier to convert between FrameMaker’s normal binary format and the formats of other systems and word processors.

opening tag

The first tag in a pair of *SGML* tags. An opening tag applies to all subsequent text up to the corresponding *closing tag*.

Part tabs

Pages used in a hard-copy book to mark the beginnings of multi-chapter sections of the book. Tabs often have shaded edges so that a reader can easily find them by looking at the edge of the book.

production editor

The person who handles production for a book—can include anything from incorporating an online book into a software product, to checking a hard-copy book’s pagination, to doing color separations for figures.

SGIDOC DTD

One of the *DTDs* used by SGI.

SGIDOCBK DTD

One of the *DTDs* used by SGI.

SGML

Standard Generalized Markup Language: a standardized format for describing structured documents in terms of their nested structures.

short name

Another name for the *short title* of a book.

short title

An abbreviated form of the title of an IRIS InSight book. Appears as an icon label when the book window is iconified. Also called a “short name.”

unknown error

An error during the *bookbuild* process with a non-obvious cause, not recognized by the build tools.

Index

A

Adobe Illustrator images, 94
.ai files, 95
anchored frames. *See* frames
applications, launching from InSight, 77
applying the template, 29
AppNum, 31
AppTitle, 31
audience for this guide, xx
autonumbering problems, 71

B

bibliography, xxi
BNF Formatting, 42, 43
Bold character tag, 61
BOOK_FILES variable, 111
BOOK_ICON variable, 111
bookbuilding
 debugging, 127-159
 diagram, 177
 help, 170
 overview, 109-125
 troubleshooting, 121
.book file, 18

book file, 65
 adding files, 66
 adding generated files, 69
 setting up files, 66
 setting up generated files, 69
booklist.txt file, 123, 175
books
 building. *See* book-building
 checking, 159
 linking between. *See* links
 titles, 20
 viewing, 124
books directory, 115
BOOKSHELF variable, 111
bsnap utility, 99, 100, 102
build process. *See* book-building
Bullet, 33
bulleted lists, 33
BulletInd, 33
BulletInd4, 33
BulletSquareInd, 33
Button character tag, 61
buttons, help, 164
.bw files, 95
BW variable, 97

C

- CalloutBold* character tag, 61
- Callout* character tag, 61
- CalloutSmall* character tag, 61
- captures, screen. *See* screen captures
- Caution*, 54
- CautionInd1*, 54
- CautionInd2*, 54
- CautionInd4*, 54
- CautionInd5*, 54
- CGM figures, 178
- ChapNum*, 31
- ChapTitle*, 31
- characters
 - formatting, 74
 - non-alphanumeric, entering, 76
- character tags, 60
 - Bold*, 61
 - Button*, 61
 - Callout*, 61
 - CalloutBold*, 61
 - CalloutSmall*, 61
 - Command*, 61
 - DocTitle*, 61
 - FileName*, 61
 - Function*, 61
 - GlossaryItem*, 61, 84
 - HardwareLabel*, 61
 - Italics*, 61
 - Keyword*, 61
 - Launchword*, 61, 91
 - Refpage*, 61, 89
 - ScreenDisplay*, 61
 - Subscript*, 62
 - Superscript*, 62
 - Symbol*, 62
 - UserInput*, 62
 - using, 74, 75
 - Variable*, 62
 - See also* tags
- Checkoff*, 43
- Checkoff1st*, 43
- chunking, 4
- clbatch* utility, 178
- Code*, 38, 39, 40
- code, 38
 - in text, 38
 - sample programs, 39
- Code1st*, 39, 40
- CodeExt*, 39, 40
- CodeExt1st*, 39, 40
- code formatting, 75
- CodeInd*, 39, 40
- CodeInd1st*, 39, 40
- CodeInd2*, 39, 40
- CodeInd2.1st*, 39, 40
- CodeInd4*, 39, 40
- CodeInd4.1st*, 39, 40
- CodeInd5*, 39, 40
- CodeInd5.1st*, 39, 40
- CodeMargin*, 39, 41
- CodeMargin1st*, 39, 41
- CodeTitle*, 41
- Command* character tag, 61
- Comment conditional text tag, 64
- commonbookdef, 20
- commonbookdefs* file, 123
- commonbookrules, 20
- commondocdef, 20
- commondocrules, 20
- conditional text
 - OnlineOnly* tag, 105, 163
 - PrintOnly* tag, 105

consistency of terminology, 5
See also naming conventions
 context sensitive help, 164, 168
 conventions. *See* naming conventions
 copyright symbol, 76
 creating links. *See* links
 cross-book links. *See* links
 cross references. *See* links

D

dashed lists, 34
 dashes, 76
DashInd, 34
 deleting files, 117
 directories
 books, 115
 figures, 95
 generated by *make*, 113
 help, 18
 online, 18, 93, 96
 orig, 18, 93, 95
 print, 18, 93, 103
 prod, 18
 tabs, 18
 working, 16
 display problems, checking for, 159
 DOCBK, 178
.doc files, 18, 178
DocTitle character tag, 61
 documenting code. *See* code

E

em dashes, 76
 en dashes, 76

.eps files, 95
Equation, 43
 Equations, 43
 equations, online, 44, 74
.err files, 128
 errors
 cross references, 137
 display problems, 159
 example, 129-132
 figures, missing, 159
 files, 128
 fixing, 142-158
 formatting, 159
 generating, 119
 help unavailable, 174
 links, broken, 159
 location of, 142
 make, 122
 messages, 128
 reports, 128, 133-141
 translation, 143
 types of, 142
 unknown, 158
 See also troubleshooting
Example, 38
ExampleInd, 38, 39
ExampleInd2, 38, 39
ExampleInd4, 38, 39
ExampleInd5, 38, 39
ExampleTitle, 41
 extensions, filename, 95
 extracting helpkeys, 173

F

Fig, 46, 98
FigTitle, 46, 98

figures, 46-47
 composite, 93, 94
 creating, 93-107
 customized print figures, 105
 directories, 95
 generating, 117
 guidelines, 106
 importing, 93
 inserting in tables, 49
 links, creating, 77
 `Makefile` editing, 96-97
 micro-adjusting, 47
 missing, 159
 naming, 94
`FileName` character tag, 61
files
 converting, 178
 error files, 128
 generated by `make`, 113
 intermediate, 114
 locked, 121, 122
 naming, 95
 removing, 117
 SGML, 128
 templates, 22
`fmbatch` utility, 178
footnotes
 creating, 76
forced return, 35
formats
 characters, 74
 frontmatter, 159
 images, 94
FrameMaker
 character designer, 74
 equations, 74
 index markers, 84
 line art, 94
 Smart Spaces, 75
 tags, using, 73-76, 163
 templates, 22

frontmatter
 formats, 159
FULL_TITLE variable, 110
`.full` file, 128, 133
full titles, 20
`Function` character tag, 61

G

generated files, 27, 65, 70
`.gif` files, 95
GIF image format, 94
GIF variable, 97
glossary
 links, creating, 77, 84
 QA report, 133, 141
`GlossaryItem` character tag, 61
GlossaryTitle, 31
guidelines for figures, 106

H

`HangingList`, 35
`HangingListInd`, 35
hanging lists, 35
hard returns, tagging, 75
`HardwareLabel` character tag, 61
`Heading1`, 31
`Heading2`, 31
`Heading3`, 31
headings
 cross referencing, 79
 guidelines, 5

-
- help, 161, 161-176
 - buttons, 164
 - context sensitive, 164, 168
 - creating, 161-176
 - helpmap file, 165
 - incorrect, 175
 - installing, 174
 - menus, 164, 168
 - missing, 175
 - missing items, 176
 - tagging, 163
 - testing, 170
 - troubleshooting, 174
 - Web pages, 170
 - help* directory, 18
 - helpkey, 173
 - helpmaps, 162
 - mapping, 162
 - widget strings, 162
 - HelpSubTopic conditional text tag, 64
 - HelpSubTopic paragraph tag, 164
 - HelpTopic*, 64
 - HelpTopic paragraph tag, 163
 - help topics, 163
 - Hint*, 51
 - HintInd1*, 51
 - HintInd2*, 51
 - HintInd4*, 51
 - HintInd5*, 51
 - Hints, 51
 - hipass3* utility, 99
- I**
- .idx* file, 115
 - Illustrator images, 94
 - images
 - capturing. *See* screen captures
 - formats
 - filename extensions, 95
 - supported, 94
 - importing, 103
 - original, 93, 95
 - unsupported, 104
 - See also* figures
 - Import Formats, 29
 - index
 - links, 84
 - links, creating, 77
 - IndexEntry conditional text tag, 64, 85
 - indexgen* utility, 178
 - index tips, 87
 - information, repeating, 7
 - INLINE_MEDIA variable, 112
 - inline media, 77, 90
 - InlineObj*, 64, 64, 90
 - InlineTitle*, 64, 64, 90
 - InSight viewer
 - checking books, 159
 - features, 2-3
 - help. *See* help
 - launching applications, 77
 - links, 77
 - Table of Contents view, 77
 - Inso, 178
 - IPAppendix.doc, 28
 - IPBook, 26
 - IPChap.doc, 28
 - IPFront.doc, 27
 - IPGlossary.doc, 29
 - IPIntro.doc, 28
 - IPIX.doc, 27
 - IPLOE.doc, 27

IPLOF.doc, 27
IPLOT.doc, 27
IPNoChap.doc, 28
IPPartTab.doc, 28
IPTab.doc, 28
IPTemplates
 applying the templates, 29
 a quick tutorial, 26
 copying files to workarea, 22
IPTOC.doc, 27
Italics character tag, 61
izoom utility, 99

K

Keyword character tag, 61

L

language support, 20
Launchword character tag, 61, 63, 91
licensing, 124
line art, 94
links
 automatic, 77
 broken, 159
 creating, 77-89
 cross-book, 63
 creating, 7, 82-83
 glossary, 84
 guidelines, 6-7
 headings, 4, 79
 index, 84
 inline media, 90
 QA report, 133, 137
 reference pages, 89
 structural, 77

 to another book, 82-83
 to applications, 90
 to URLs, 90
 within a book, 79
List, 34
List1st, 34
ListInd, 34
ListInd1st, 34
ListInd4, 35
ListInd4.1st, 35
lists, 32-35
 bulleted, 33
 dashed, 34
 hanging, 35
 ordered, 34
 rules for use, 32
 square bulleted, 33
 tabular, 36
local working directory, 16
locked files, 121

M

make
 commands
 choosing, 118
 list of, 116
make, 11, 109, 113-121
 _online, 121
 _print, 121
 arguments, 116
 book, 120
 book.full, 118
 building an entire book, 116
 disk full, 121
 error files, 119
 errors, 122
 failures, 121

make (continued)
 figures, 117
 final, 120
 generated files and directories, 113
 initial, 118
 installing, 11
 prerequisites, 116
 removing files, 117

Makefile
 choosing a *Makefile* template, 22
 copying file to workarea, 22
 description, 18
 editing, 110
 figures, 96-97
 image files in, 93
 Makefile_sgidocbk, 20
 Makefile_sgml doc, 20
 renaming, 121
 syntax, 121

mathematical equations, 44

media, inline, 77, 90

menus
 capturing, 102
 help, 164
 rollover, 168

Messages, 42

mif2sgml, 178

MIF files, 114, 178

miftocgm utility, 178

mksgidoc utility, 178

Msg, 42

Msg1st, 42

MsgExpl, 42

MsgMargin, 43

MsgMargin1st, 43

N

naming conventions
 books, 20
 figures, 94
 files, 19

non-alphanumeric characters, 76

Note, 53

NoteInd1, 53

NoteInd2, 53

NoteInd4, 53

NoteInd5, 53

Notes, 53

O

online
 help. *See* help
 images, 93
 wording, 6
 writing tips, 4

online directory, 18, 93, 96

OnlineOnly conditional text tag, 44, 64, 90, 105, 163

ordered lists, 34

orig directory, 18, 93, 95

P

paragraph tags
 Bullet, 33
 BulletInd, 33
 BulletInd4, 33
 BulletSquareInd, 33
 Caution, 54
 CautionInd1, 54
 CautionInd2, 54
 CautionInd4, 54

paragraph tags (*continued*)

CautionInd5, 54
Checkoff, 43
Checkoff1st, 43
Code, 39, 40
Code1st, 39, 40
CodeExt, 39, 40
CodeExt1st, 39, 40
CodeInd, 39, 40
CodeInd1st, 39, 40
CodeInd2, 39, 40
CodeInd2.1st, 39, 40
CodeInd4, 39, 40
CodeInd4.1st, 39, 40
CodeInd5, 39, 40
CodeInd5.1st, 39, 40
CodeMargin, 39, 41
CodeMargin1st, 39, 41
CodeTitle, 41
DashInd, 34
Equation, 43
Example, 38
ExampleInd, 38, 39
ExampleInd2, 38, 39
ExampleInd4, 38, 39
ExampleInd5, 38, 39
ExampleTitle, 41
Fig, 46, 98
FigTitle, 46, 98
HangingList, 35
HangingListInd, 35
Heading1, 31
Heading2, 31
Heading3, 31
HelpSubTopic, 164
HelpTopic, 64, 163
Hint, 51
HintInd1, 51
HintInd2, 51
HintInd4, 51
HintInd5, 51
InlineObj, 64, 90
InlineTitle, 64, 64, 90
List, 34
List1st, 34
ListInd, 34
ListInd1st, 34
ListInd4, 35
ListInd4.1st, 35
Msg, 42
Msg1st, 42
MsgExpl, 42
MsgMargin, 43
MsgMargin1st, 43
naming scheme, 31
Note, 53
NoteInd1, 53
NoteInd2, 53
NoteInd4, 53
NoteInd5, 53
Shortcut, 57
ShortcutInd, 57
TableHead, 47
TableText, 47
TableTitle, 47
TabularListText, 36
Text, 37
TextInd, 37
TextInd2, 37
TextInd4, 37
TextInd5, 37
Tip, 52
TipInd1, 52
TipInd2, 52
TipInd4, 52
TipInd5, 52
Warning, 55
WarningInd1, 55
WarningInd2, 55
See also tags
PBMPlus utilities, 178

PostScript
 images, 94
 variable, 97
PRINT_BW variable, 96
PRINT_COLOR variable, 96
print directory, 18, 93, 103, 105
PrintOnly conditional text tag, 64, 105
print-oriented wording, 6
production editor, xx
.ps files, 95
pull-down menus, capturing, 102

Q

QA reports, 133, 137
quotation marks, smart, 75

R

README file, 18
reference pages, links to, 77
references, xxi
Refpage character tag, 61
registered trademark symbol, 76
removing files, 117
repeating information, 7
requirements, software, 9
returns, soft, tagging, 75
.rgb files, 95, 100
RGB image format, 94
RGB variable, 97
rollover menus, 168

S

sample programs, 39
screen captures, 74, 99-102
 backgrounds, 100
 equations, 74
 formats, 99
 menus, 102
 naming, 100
 tools, 99
 windows, 100
ScreenDisplay character tag, 61
screen snaps. *See* screen captures
scrsave utility, 99
SGIAUDIO, 64, 90
SGIDOCBK DTD, 22, 109, 178
SGIDOC DTD, 22, 109, 178
SGI Help. *See* help
SGIINVENTOR, 64, 90
SGIRGB, 64, 90
SGIVIDEO, 64, 90
SGML
 description, 128
 files, 18, 114, 128, 178
 structure, 74, 129
 tags, 128
 tag structure, 133
short_title.full file, 128
Shortcut, 57
ShortcutInd, 57
shortcuts, 57
short titles, 21
Smart Quotes, 75
Smart Spaces, 38, 75
snaps, screen. *See* screen captures
snapshot tool, 74, 101
soft returns, tagging, 75

- software requirements, 9
- spaces in code, 75
- square bulleted lists, 33
- ssnap* utility, 99
- structural links, 77
- style, online, 4
- Subscript* character tag, 62
- Superscript* character tag, 62
- supplementary reading, xxi
- Symbol* character tag, 45, 62
- symbols
 - copyright, 76
 - non-alphanumeric, entering, 76
 - trademark, 76

T

- Table Continuation* variable, 50
- table formats
 - TabularList*, 36
- TableHead*, 47
- tables, 47-50
 - creating, 47
 - customizing, 48
 - footnotes, 76
 - links, creating, 77
- TableText*, 47
- TableTitle*, 47
- tabs
 - in code, 75
- tabs* directory, 18
- TabularList*, 36
- tabular lists, 36
- TabularListText*, 36

- tags
 - character. *See* character tags
 - creating, 74
 - paragraph. *See* paragraph tags
 - sequences, valid, 74
 - SGML, 133
 - tips, 73-76
 - variables, 145
- templates
 - annotated, 129
 - obtaining, 22
 - structure, 74
 - tags, using, 73-76
- terminology, consistency of, 5
- testing online help, 170
- Text*, 37
- TextInd*, 37
- TextInd2*, 37
- TextInd4*, 37
- TextInd5*, 37
- Tip*, 52
- TipInd1*, 52
- TipInd2*, 52
- TipInd4*, 52
- TipInd5*, 52
- Tips, 52
- tips for online writing, 4
- titles, 20
- TITLE variable, 110
- tools, setting up, 10
- trademark symbol, 76
- translation errors, 135, 143

troubleshooting
 autonumbering problems, 71
 bookbuilding, 127-159
 displaying, 159
 glossary links, 84
 help, online, 174
 make, 121

U

unknown errors, 158
URLs, 170
UserInput character tag, 62

V

Variable character tag, 62
variables, character tags, 145
VERSION variable, 111
viewing a book, 124

W

Warning, 55
WarningInd1, 55
WarningInd2, 55
Warnings, 55
widgets, 168
widget strings, 162
windows, capturing, 100
winsnap utility, 99
wording, print-oriented, 6
working directory, 16
World Wide Web, 170
writing tips for online, 4
WWW, 170

X

.xwd files, 95
XWD image format, 94
XWD variable, 97

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-2863-002.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
 - On the Internet: techpubs@sgi.com
 - For UUCP mail (through any backbone site): *[your_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
1600 Amphitheatre Parkway, M/S 535
Mountain View, California 94043-1351