Apex Content Solutions

Knowledge Solutions, Global Reach

Gaining Control Over Help-Specific Features with a FrameMaker EDD

Presented by Martin R. Smith
Senior Consultant
Apex CoVantage

WebWorks ePublisher Pro is an outstanding tool for producing on-line documentation from documents created with Adobe FrameMaker, Microsoft Word, and most recently DITA Content.

Creating an on-line help system using WebWorks ePublisher Pro involves filling out a style sheet that maps paragraph and character styles used in the source documents with the styles displayed in the help system. Assuming writers use styles consistently, the same ePublisher template can be used to consistently produce on-line help systems for every product a technical writing team handles.

However, the ePublisher template does more than format the styles used in the source documents for on-line delivery. The style sheet also determines help-specific behavior, such as the distinction between topics and mid-topics, and the creation of pop-up windows, and sections that expand and collapse. Handling these help-specific features can present some challenges when developing document and ePublisher templates that align with one another. For example, consider the relation between sections in a document and the topics and mid-topics used in a help system.

A manual is likely to include one or more chapters, and three to four levels of headings. The paragraph style names used to format the chapter title and headings in the source documents could include the following:

- ChapterTitle
- Heading 1
- Heading 2
- Heading 3

In the corresponding ePublisher template, paragraph styles Chapter Title, Heading 1, and Heading 2 are likely to become new topics in the help system. Heading 3 paragraphs are likely to become mid-topics in the help system. If your projects require more flexibility over topic / mid-topic breaks you have two choices: preview the help system in ePublisher and modify topic / mid-topic breaks on a case-by-case basis, or add additional paragraph styles to the document. To control topic / mid-topic breaks using paragraph styles we need to modify our original list of paragraph styles to include the following:

- Chapter Title
- Heading 1, Topic
- Heading 1, MidTopic
- Heading 2, Topic
- Heading 2, MidTopic
- Heading 3, Topic
- Heading 3, MidTopic

Adding paragraph styles to control additional help-specific features such as drop-down sections and pop-ups further expands the list of paragraph styles to include the following:

- Chapter Title
- Heading 1, Topic
- Heading 1, MidTopic
- Heading 1, Dropdown, Start Closed
- Heading 1, Dropdown, Start Open
- Heading 1, PopUp, Include Text
- Heading 1, PopUp, Exclude Text
- Heading 2, Topic
- Heading 2, MidTopic
- Heading 2, Dropdown, Start Closed
- Heading 2, Dropdown, Start Open
- Heading 2, PopUp, Include Text
- Heading 2, PopUp, Exclude Text
- Heading 3, Topic
- Heading 3, MidTopic
- Heading 3, Dropdown, Start Closed
- Heading 3, Dropdown, Start Open
- Heading 3, PopUp, Include Text
- Heading 3, PopUp, Exclude Text

As the number of paragraph styles multiplies, the impracticality of this approach becomes clear. A writer must choose from one of six paragraph styles to format a first level heading. All six variations of the heading would be formatted identically in the source document, with the only distinction being the name of the paragraph style used to format the text. For this reason, ePublisher provides the Document Designer, which allows for the inclusion of topic / mid-topic breaks and help-specific features on a case-by-case basis.

Working with the Document Designer to change topic / mid-topic breaks and add help-specific features adds an additional step in the document production workflow, as changes are made for the benefit of the on-line help system after the final text is complete and ready for production. In addition, any changes made using the Document Designer are not visible in the source document, which requires writers to revisit the design of the help system when working on the next release of the documentation. In order to achieve a true, single-source-publishing workflow where all help-specific features are defined in the source file, there needs to be a way to format documents for print independent of the paragraph styles used to produce the help system. For Adobe FrameMaker users, this is a possibility.

This paper describes how to create a single-source-publishing workflow built around a customized FrameMaker EDD (element definition document) and a WebWorks ePublisher Pro template. For those unaccustomed to structured authoring, an EDD contains a set of rules that format text for print based on context. Paragraph tags are not used to format text in structured documents, which frees them for use to drive the help system.

The emphasis here is on creating a minimalist EDD for producing printed documents and on-line help systems. For those familiar with the hundreds of elements defined for industry-standard EDDs such as DocBook, the EDD presented here uses only nine content elements.

Using the Sample EDD to Create Printed and On-Line Documentation

This section describes how to use the sample page template and underlying EDD to produce printed documents and on-line help system. Topics include the following:

- Running Adobe FrameMaker in Structured Mode
- Creating a New Structured Document
- Opening the Element Catalog and Structure View
- Filtering the Elements Displayed in the Element Catalog
- Inserting the Root Element
- Inserting the Title Element
- Inserting the First Para Element
- Adding Additional Content Elements
- Adding Section Elements
- Controlling Topic and Mid-Topic Breaks
- Introduction to DropDown Sections and Pop-Up Windows
- Creating DropDown Sections
- Creating Pop-Up Windows
- Adding Context IDs for Context-Sensitive Help
- Assigning Custom Page Styles
- Using the Alert Element

Running Adobe FrameMaker in Structured Mode

Adobe FrameMaker 7.x runs in two modes: unstructured mode and structured mode. You must configure FrameMaker to run in structured mode before you can work with the sample files provided with this paper.

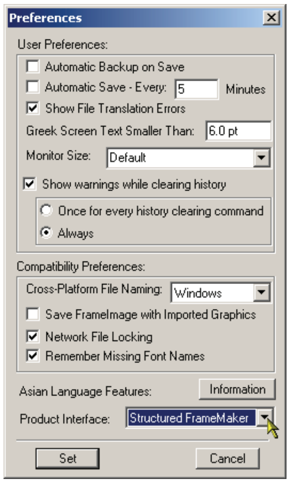
To configure FrameMaker to run in structured mode, do the following:

1. Run Adobe FrameMaker.
2. Select Preferences ▶ General… from the File menu.

   The Preferences menu will open.
3. Ensure that Structured FrameMaker is selected in the Product Interface menu, as shown in Figure 1.
4. Exit and then restart FrameMaker.

   FrameMaker will now be running in structured mode.

Structured mode adds additional tools to FrameMaker's user interface that allow you to work with both structured and unstructured documents.



☐ Automatic Backup on Save
☐ Automatic Save - Every: 5 Minutes
☑ Show File Translation Errors
Greek Screen Text Smaller Than: 6.0 pt
Monitor Size: Default
☑ Show warnings while clearing history
○ Once for every history clearing command
● Always
Cross-Platform File Naming: Windows
☐ Save FrameImage with Imported Graphics
☑ Network File Locking
☑ Remember Missing Font Names
Asian Language Features:
Product Interface: Structured FrameMaker

Figure 1: Configuring FrameMaker to run in structured mode

Creating a New Structured Document

To create a new FrameMaker document based on the sample page template and EDD, do the following:

1. Create a new directory for your project.
2. Copy the document named `SampleChapter.fm` from the sample CD to your project directory.
3. Rename `SampleChapter.fm` to something more descriptive.
4. Double-click to open the document in Adobe FrameMaker.

Opening the Element Catalog and Structure View

The first thing you need to do when your new document opens is configure your work environment. We will not be using the paragraph and character catalog windows to create structured documents. Instead we will be working with FrameMaker's Element Catalog and Structure View.

To open the Element Catalog and Structure View, do the following:

1. Click the [] icon in the upper right corner of the document window to open the Element Catalog.
2. Click the ☰ icon in the upper right corner of the document window to open the Structure Window.
3. Arrange the document window, Element Catalog, and Structure view on your desktop so you can work with all three of these items easily. (Refer to Figure 3.)

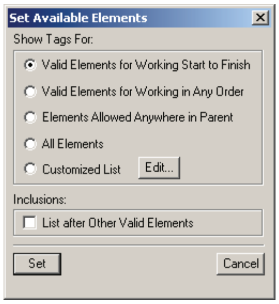
Filtering the Elements Displayed in the Element Catalog

The Element Catalog can display all of the elements defined in the EDD, or it can filter the list of elements. When it filters the list of elements, the Element Catalog displays only those elements that are valid at the current location in the document. For the purpose of this paper, we will filter the list of elements.

To only display the valid elements at a given location, do the following:

1. Click the Options button at the bottom of the Elements catalog window.

   The Set Available Elements Window will open.
2. Select Valid Elements for working Start to Finish, as shown in Figure 2.
3. Click Set to save your changes and close the Set Available Elements window.



Set Available Elements

Show Tags For:

● Valid Elements for Working Start to Finish
○ Valid Elements for Working in Any Order
○ Elements Allowed Anywhere in Parent
○ All Elements
○ Customized List

Inclusions:

☐ List after Other Valid Elements

Figure 2: Set Available Elements Window

Inserting the Root Element

With a new document open on the screen and our work space organized, we are ready to begin inserting elements and writing content.

A structured document is organized as a hierarchy of elements. Some elements contain content; other elements to not. Elements that do not contain content contain other elements. We refer to these as container elements. Every structured document begins with a special container element—called the root element—that contains all of the subsequent elements in the document. The sample page template and EDD provided with this paper contains three root elements:

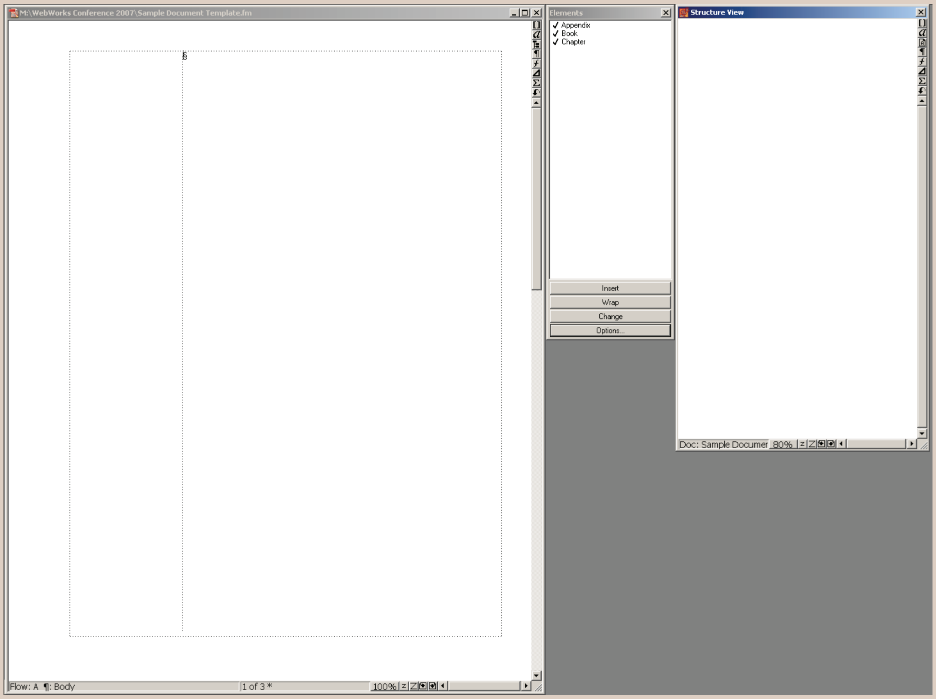
- Chapter
- Appendix
- Book

The Chapter and Appendix elements are used to begin FrameMaker documents. The Book element is used to organize the documents in a FrameMaker book file. Thus the first element we need to add to our sample document is a Chapter or Appendix element. For the purpose of this paper, we will

Figure 3: Document Window, Element Catalog and Structure View

This figure shows the new document window, based on the supplied page template and EDD (Element Definition Document).

The Element Catalog lists the three elements that are valid at the root of a document: Chapter, Appendix, and Book. Only the Chapter and Appendix elements are applicable to documents. The Book element is used to structure a FrameMaker book file.



insert a Chapter element. All of the elements we add subsequently will be contained within this Chapter element.

To insert the first element in a structured document, do the following.

1. Ensure that the Element Catalog and Structure View windows are open and that your work space is arranged as shown in Figure 3.
2. Double click on the element named Chapter in the Elements catalog window.

FrameMaker will insert a Chapter element in the document. FrameMaker will also automatically insert a ChaptNum element, as the first element contained within the Chapter element. This happens because the underlying EDD is programmed to automatically insert a ChaptNum element when you insert a Chapter element.

The Chapter and ChaptNum elements will appear in the structure view, as shown in Figure 4. The content of the ChaptNum element appears in the document as the numeral one, formatted as the chapter number in our document.

Note that the chapter and appendix numbering for the provided page template and EDD is controlled by the book file. Thus chapters and appendices will always begin with the number one until you create a book file containing these documents.

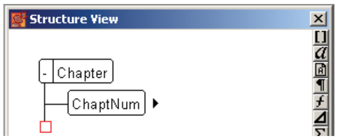


Figure 4: Chapter and ChaptNum elements in the structure view

Once the Chapter and ChaptNum elements have been added to the document, we are ready to write the title of the chapter.

Inserting the Title Element

The underlying EDD that has been imported into the page template supplied with this paper is designed to format the text of the FrameMaker document and assign paragraph styles for the benefit of WebWorks. As we can se, the underlying EDD also controls the organization of any document you create.

This particular EDD has been designed to be highly flexible, though it does impose rules on how a document is organized. When designing an EDD we can decide how strictly or loosely the EDD enforces the organization of a document. This particular EDD requires a chapter to start with the following sequence of elements:

- Chapter
  - ChaptNum
  - Title
  - Para

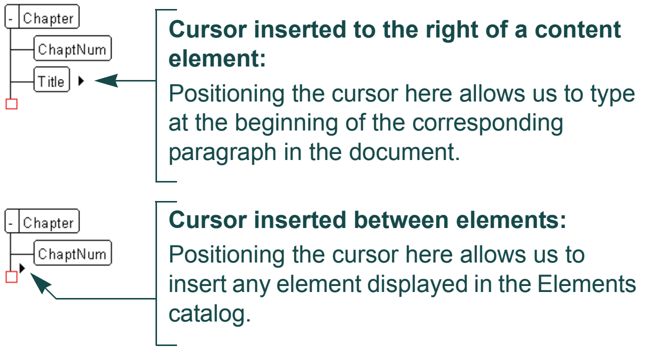
Once a chapter contains this sequence of elements, it may then contain nearly any combination of the following:

- Numbered Lists
- Bulleted Lists
- Figures
- Tables
- Sections

When we generate a help system, the chapter title always becomes a new topic in the help system. Sections within the chapter may become topics or mid-topics, at the writer's discretion.

Before we can insert the chapter title we need to understand how the cursor behaves inside the Structure View window (Figure 5):

- You can always place the cursor between the elements in the structure view. In addition, you can always place the cursor between the last element in the structure view and the red box that represents the end of the element hierarchy.
- You can only place the cursor to the right of a content element in the structure view. Placing the cursor to the right of a content element allows you to begin typing at the beginning of the corresponding paragraph in the document.
- You cannot place the cursor to the right of container elements in the structure view because container elements may only contain other elements. Container elements do not have a corresponding paragraph in the document.



Cursor inserted to the right of a content element:

Positioning the cursor here allows us to type at the beginning of the corresponding paragraph in the document.

Cursor inserted between elements:

Positioning the cursor here allows us to insert any element displayed in the Elements catalog.

Figure 5: Positioning the cursor in the structure view

To insert the Title element in the Structure View and type the title of the chapter, do the following:

1. Position the cursor beneath the ChaptNum element in the structure view and click the mouse button.
2. Double-click the Title element in the Elements catalog window.

   Because the Title element is programmed to behave as a content element in the underlying EDD, FrameMaker will insert a new paragraph in the document to contain the chapter title.
3. Type the text for the chapter title.

   Notice that the underlying EDD formats the text automatically. The EDD also assigns a paragraph tag to the title (`Topic_L1`) that will create a new topic when we import the chapter into WebWorks.

Inserting the First Para Element

The underlying EDD requires us to insert a paragraph element after the chapter title. This requirement is an editorial decision that was made during the development of the underlying EDD. In the final section of this paper, we will explore how to modify the supplied EDD, as well as how to develop an EDD of your own. We could easily modify the EDD to allow a chapter to begin with a numbered or bulleted list, figure, table, or section.

To insert the first paragraph in the chapter, do the following:

1. Position the cursor beneath the Title element in the Structure View.
2. Double-click the Para element in the Elements catalog.

   Framemaker will insert a new paragraph in the document and format the text automatically.
3. Type some text for the first paragraph.

If you press the Enter key on the keyboard, FrameMaker will automatically insert a new Para element into the structure view and add a new blank paragraph to the document.

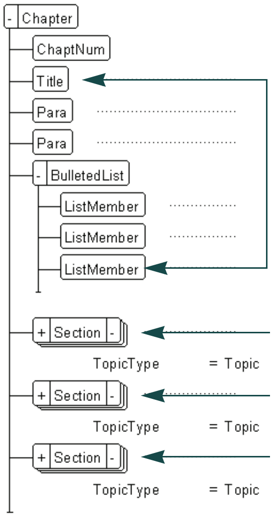
Adding Additional Content Elements

Having inserted the initial series of required elements—Chapter, ChaptNum, Title, Para—the EDD allows us to insert the following elements:

- Para
- BulletedList
- NumberedList
- Figure
- Table
- Section

You can combine the elements in this list in any order, with the exception of the Section element. A section element can only be followed by another section element. Section elements may become topics or mid-topics in the resulting help system, at the discretion of the author. Figure 6 shows how a typical chapter is organized in the structure view.

Assuming the first section is configured to become a new topic, the content of all elements between the Title and the first section will become a first-level topic in the resulting help system.



Level 1 Topic:
The content between the Title element and the last member in the list will become the content of a new, level one topic in the resulting help system.

Three Level 2 Topics:
The content of each of these three sections is configured to become a new topic in the help system.

Figure 6: Initial structure of a sample chapter

Adding Section Elements

Section elements are used to organize the content of a chapter or appendix into manageable pieces and correspond with the headings in an unstructured document. Section elements inserted following the Chapter or Appendix element correspond with the Heading 1 paragraphs in an unstructured document. Section elements can also be nested inside one another, which provides additional levels of headings: Heading 2, Heading 3, and so on. The content of a section element may also become a new topic or a mid-topic in the resulting help system.

You can expand and collapse the section (and other container elements) in the structure view by clicking the expand / collapse icons on the left side of each element. Section elements also have an expand / collapse icon on the right side, which expands and collapses the element's attributes (Figure 7). In the next section we will describe how to use attributes to control topic / mid-topic breaks in the resulting help system.

Controlling Topic and Mid-Topic Breaks

The underlying EDD in the sample template implements topic / mid-topic breaks using a feature called an attribute. An element in a FrameMaker document may contain one or more attributes. An attribute is like a variable in that it can store a value. The underlying EDD can format the text differently, depending on the value assigned to element attributes.

When you insert a section element in the sample template, FrameMaker will display a pop-up window (Figure 8), prompting you to set the value of the TopicType attribute. Your choices include Topic or MidTopic.

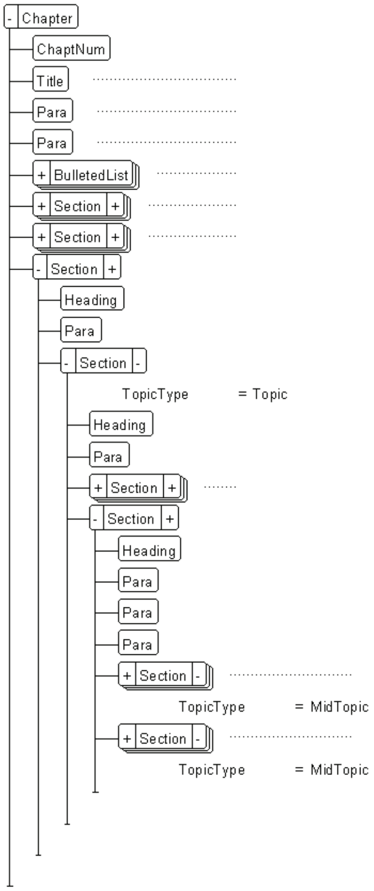
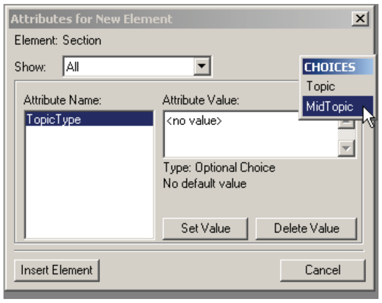


Figure 7: Expanding and collapsing elements and element attributes in the structure view



Element: Section

Show: All

Attribute Name: TopicType

Attribute Value: <no value>

Type: Optional Choice
No default value

Figure 8: Setting the TopicType attribute after inserting a new section element

Selecting Topic or MidTopic has no affect on how FrameMaker formats the document for print. However, FrameMaker will assign a different paragraph tag to the heading's paragraph, depending on the value you select. FrameMaker will automatically apply one of five paragraph tags to the heading depending on the nesting level of the Section element and the value of the TopicType attribute. These paragraph tags include the following:

- Topic_L1
- Topic_L2
- Topic_L3
- Topic_L4
- Topic_L5
- Topic_Mid

The WebWorks Template provided with this paper contains paragraph styles that align with the paragraph styles automatically assigned by FrameMaker.

You can change the topic / mid-topic designation of a section at any time by changing the value of the TopicType value. To change a new topic into a mid-topic (or a mid-topic into a topic) do the following:

1. Locate the section you want to change in the structure view.
2. If necessary, expand the element so you can view the Double-click on the TopicType attribute.

   FrameMaker will display the Attributes window.
3. Select the TopicType attribute in the Attributes window and select Topic or MidTopic from the Choices drop-down menu.
4. Click the Set Value button to make the change.
5. Click the Done button to close the Attributes window.

   FrameMaker will change the paragraph tag assigned to the heading paragraph when you click the Done button.

Introduction to DropDown Sections and Pop-Up Windows

WebWorks ePublisher Pro allows you to create drop-down sections that behave in two different ways:

- DropDowns that the user expands by clicking on the heading
- DropDowns that the user collapses by clicking on the heading

WebWorks ePublisher Pro also offers two variations of PopUp windows:

- PopUp windows with content that is also displayed in one of the help system topics
- PopUp windows with content that is not available anywhere else in the help system

Creating drop-downs and pop-ups within the source files presents a unique set of challenges for help authors. Not only do you need to use special paragraph tags to mark the beginning of the drop-down or pop-up, you need to tag all of the subsequent paragraphs that get included in the resulting drop-down or pop-up window. The need to apply a designated paragraph tag to the member paragraphs has the potential of tripling the number of paragraph styles used to format the text. This is where the enclosed page template and EDD really makes a difference.

The included page template and underlying EDD contains an element named SubSection. When you insert a SubSection element, FrameMaker will prompt you to set the TopicType attribute. Choices for the TopicType attribute include the following:

- DropDown: Start Closed
- DropDown: Start Open
- PopUp: Include Text
- PopUp: Exclude Text

Just like a Section element, a SubSection element must be followed by a heading and an initial paragraph. Once you have inserted this initial series of elements, you are free to choose from these additional types of elements:

- Paragraphs
- Bulleted Lists
- Numbered Lists
- Alerts

As you add content to the SubSection element, FrameMaker automatically applies the correct paragraph tags to all of the paragraphs contained within the drop-down section or pop-up. This is possible because of the hierarchical nature of a structured document. Because all of the elements in the drop-down or pop-up are wrapped inside the SubSection element, the underlying EDD is able to detect where these help-specific features begin and end in the FrameMaker source files and automatically assign the correct paragraph tags.

Creating DropDown Sections

The sample page template and underlying EDD includes the SubSection element that can be used to create drop-down sections. The underlying EDD automatically assigns the correct paragraph tags necessary for WebWorks to begin the drop-down section. The EDD also assigns the paragraph tags necessary for WebWorks to include all subsequent paragraphs in the SubSection in the drop-down section.

To insert a SubSection element that will become a drop-down section in the resulting help system, do the following:

1. Using the structure view, navigate to the location in your document where you would like to begin a drop-down section.
2. Double-click on the SubSection element in the Elements catalog.

   FrameMaker will prompt you to select the TopicType.
3. Select DropDown: Start Closed or DropDown: Start Open and click the Insert Element button.
   - Select DropDown: Start Closed to create a drop-down that is initially collapsed by the help system.
   - Select DropDown: Start Open to create a drop-down that is initially expanded by the help system.
4. Type the text for the drop-down heading.
5. Insert a paragraph element and type the initial paragraph of the drop-down.
6. Continue adding text to the drop-down, as necessary. A dropdown may contain any assortment of paragraphs, bulleted lists, numbered lists, and alerts.

   All of the content inside the SubSection element will become part of the drop-down when you generate the help system.

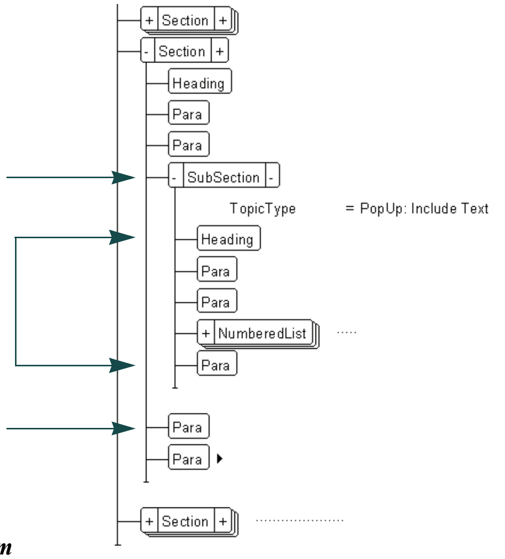
To end the drop-down section do the following:

1. Collapse the SubSection element in the structure view.
2. Insert the next element after the DropDown element.

   Only the elements contained within the SubSection element will be included in the drop-down.

Creating Pop-Up Windows

The SubSection element can also be used to create pop-up windows. WebWorks allows you to create two types of pop-up windows: pop-up windows that exclude text from a topic and pop-up windows that duplicate text from a topic.



DropDown:
Start Closed:

The TopicType attribute of this SubSection element is set to PopUp: Include Text.

The text of the Heading element will become a collapsed, drop-down heading in the resulting help system. The text of all subsequent elements in the SubSection element will expand when a user clicks on the drop-down in the help system.

This Para element is not nested inside of the SubSection element and therefore its text does not get folded into the pop-up in the resulting help system.

Figure 9: Using a SubSection element to create a drop-down in the resulting help system

To create a pop-up window in the help system, do the following:

1. Using the structure view, navigate to the location in your document where you would like to create a pop-up window.
2. Double-click on the SubSection element in the Elements catalog.

   FrameMaker will prompt you to select the TopicType.
3. Select PopUp: Include Text or PopUp: Exclude Text and click the Insert Element button.
   - Select PopUp: Exclude Text to create a pop-up that excludes text from the current topic. The only way for the help system to display this text, is to create a cross reference to the Heading element at the beginning of the DropDown element.
   - Select PopUp: Include Text to create a pop-up that contains text from the current topic. In order to display the text as a Pop-Up you need to insert a cross reference to the heading at the beginning of the DropDown element.
4. Type the text for the Pop-Up heading.
5. Insert a paragraph element and type the initial paragraph of the pop-up.
6. Continue adding text to the drop-down, as necessary. A dropdown may contain any assortment of paragraphs, bulleted lists, numbered lists, and alerts.

   All of the content inside the SubSection element will become part of the drop-down when you generate the help system.

To end the drop-down section do the following:

1. Collapse the SubSection element in the structure view.
2. Insert the next element after the DropDown element.

   Only the elements contained within the SubSection element will be included in the drop-down.

Adding Context IDs for Context-Sensitive Help

Implementing context-sensitive help with WebWorks ePublisher Pro involves inserting markers into the appropriate headings in the FrameMaker source files. Each marker, typically named TopicAlias, must contain a topic alias that aligns with a topic ID in the software application.

The sample page template and underlying EDD implement context-sensitive help with a special element named TopicAlias. The TopicAlias element inserts a marker at the end of the paragraph associated with a section's Heading element. Figure 10 shows the TopicAlias element in the structure view.

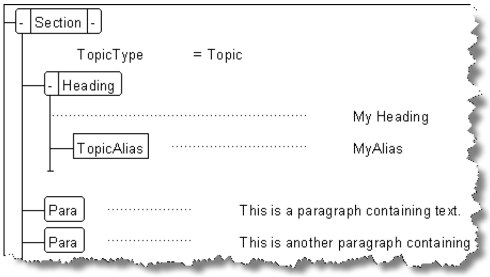
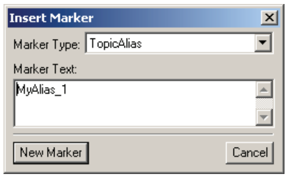

Section
TopicType = Topic
Heading
My Heading
TopicAlias
MyAlias
Para
This is a paragraph containing text.
Para
This is another paragraph containing

Figure 10: Adding a TopicAlias to a section

To insert a context ID for a help topic, do the following:

1. Using the structure view, navigate to the section containing the topic you want to link to your software application.
2. Expand the Section to reveal he Heading element.
3. Click to the right of the text for the Heading element displayed in the Structure View.
4. Double-click the TopicAlias element in the Elements catalog.

   The Insert Maker window will open.



Marker Type: TopicAlias

Marker Text: MyAlias_1

5. Type the topic alias in the Insert Marker window and click the New Marker button.

   The Insert Marker window will close and FrameMaker will display the topic alias in the Structure View.

Assigning Custom Page Styles

Some help systems use custom page styles with background images to denote major transition points in the documentation. The sample FrameMaker page template and underlying EDD allows you to apply custom page styles for the highest level topic in each chapter. The EDD implements custom page styles with a special element named PageStyle. The PageStyle element inserts a marker at the beginning of the paragraph associated with a Chapter or Appendix Title element. Figure 11 shows the PageStyle element in the structure view.

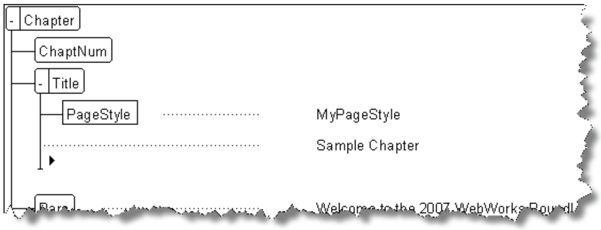
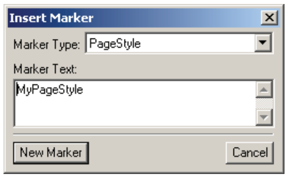


Figure 11: Applying a custom page style

To apply a custom page style to the highest level topic in a chapter, do the following:

1. Using the Structure View, expand the highest level element in the current chapter or appendix.
2. Click to the left of the text for the Title element displayed in the Structure View.
3. Double-click the PageStyle element in the Elements catalog.

   The Insert Marker window will open.



Marker Type: PageStyle

Marker Text: MyPageStyle

4. Type the name of a page style defined in your WebWorks ePublisher Pro template and click the New Marker button.

   The Insert Marker window will close and FrameMaker will display the PageStyle marker in the Structure View.

If you would like the option of selecting custom page styles for every topic, you can easily modify the EDD to allow PageStyle marker elements to occur in-line with Section Heading elements.

Using the Alert Element

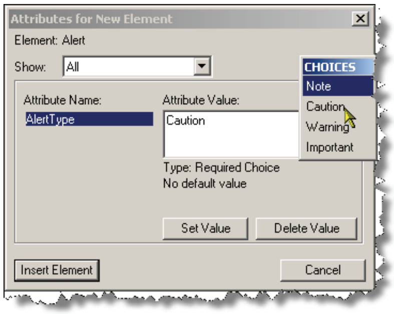
In keeping with the minimalist philosophy behind the sample page template and EDD, a single element named Alert is used to create Note, Caution, Warning, and Important messages. An attribute of the Alert element allows you to select from a list of available message types. An alert may contain one or more paragraphs. The EDD formats alerts containing more than one paragraph as a bulleted list. The EDD automatically applies reference page graphics above and below the first and last paragraph of the alert. Figure 12 shows an example of a Note and Caution statement in a FrameMaker document.

To insert an alert statement, do the following:

1. Using the Structure View, navigate to the Section where you would like to insert an Alert statement.

   The EDD allows Alerts to occur anywhere within a section, following the initial Heading and Para elements.
2. Double click on the Alert element in the Elements catalog.

   FrameMaker will prompt you to select the alert type.



Element: Alert

Show: All

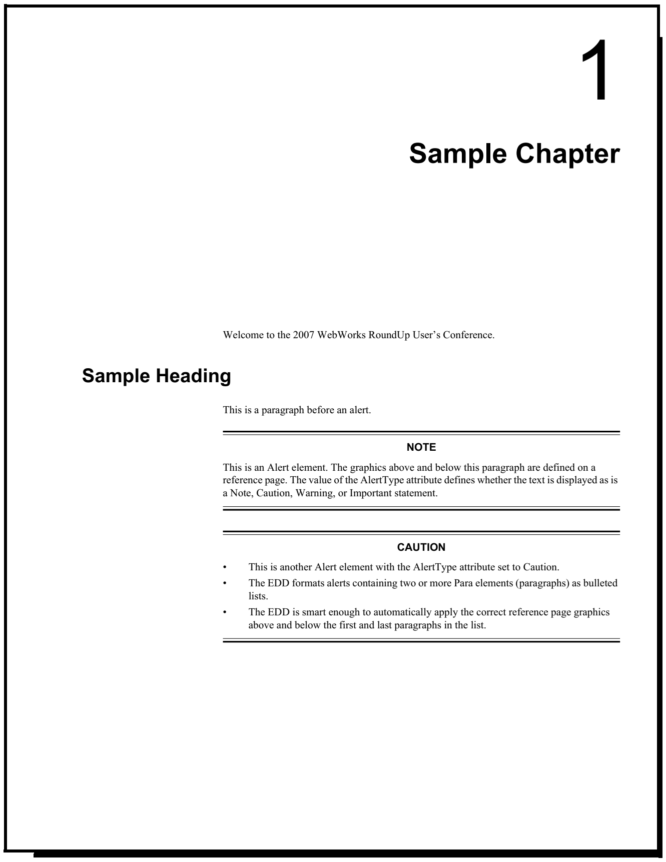
Attribute Name: AlertType

Attribute Value: Caution

Type: Required Choice
No default value

3. Select Note, Caution, Warning, or Important and click the Insert Element button.

   FrameMaker will automatically add the first Para element to the Alert and you can begin typing. If you press Enter and type a second paragraph in the Alert, FrameMaker will automatically format the paragraphs of the Alert as a bulleted list.



1

Sample Chapter

Welcome to the 2007 WebWorks RoundUp User's Conference.

Sample Heading

This is a paragraph before an alert.

NOTE

This is an Alert element. The graphics above and below this paragraph are defined on a reference page. The value of the AlertType attribute defines whether the text is displayed as is a Note, Caution, Warning, or Important statement.

CAUTION

- This is another Alert element with the AlertType attribute set to Caution.
- The EDD formats alerts containing two or more Para elements (paragraphs) as bulleted lists.
- The EDD is smart enough to automatically apply the correct reference page graphics above and below the first and last paragraphs in the list.

Figure 12: Sample document with note and caution statements

Exploring the Interaction between Elements, Paragraph Styles, and WebWorks

The sample CD contains a WebWorks ePublisher Pro template that aligns with the sample FrameMaker page template and underlying EDD.

Table 1 lists the content elements defined in the EDD and all of the paragraph styles defined in the WebWorks template. The table distinguishes between the WebWorks paragraph styles used to create paragraphs, drop-down sections, and pop-ups.

The underlying EDD assigns all of these paragraph styles automatically, based on context and attributes selected by the author.

Table 1: FrameMaker Elements and WebWorks Paragraph Styles

Content Elements in the FrameMaker EDD	WebWorks Paragraph Styles		
	Paragraphs	Drop Down Sections	Pop Up Sections
ChaptNum	WW_Omitted		
AppNum	WW_Omitted		
Title	Topic_L1		
Heading	Topic_L2 Topic_L3 Topic_L4 Topic_L5 Topic_Mid	DropDown_Open DropDown_Closed	PopUp_Include PopUp_Exclude
ListMember	Bullet_L1 Bullet_L2 Bullet_L3 Numbered_L1 Numbered_L2 Numbered_L3 Para_L1 Para_L2 Para_L3	DD_Bullet_L1 DD_Bullet_L2 DD_Bullet_L3 DD_Numbered_L1 DD_Numbered_L2 DD_Numbered_L3 DD_Para_L1 DD_Para_L2 DD_Para_L3	PopUp_Bullet_L1 PopUp_Bullet_L2 PopUp_Bullet_L3 PopUp_Numbered_L1 PopUp_Numbered_L2 PopUp_Numbered_L3 PopUp_Para_L1 PopUp_Para_L2 PopUp_Para_L3
Para	Alert_Note_First Alert_Note_Only Alert_Important_First Alert_Important_Only Alert_Caution_First Alert_Caution_Only Alert_Warning_First Alert_Warning_Only Alert_Last Alert_Mid Para	DD_Alert_Note_First DD_Alert_Note_Only DD_Alert_Important_First DD_Alert_Important_Only DD_Alert_Caution_First DD_Alert_Caution_Only DD_Alert_Warning_First DD_Alert_Warning_Only DD_Alert_Last DD_Alert_Mid DD_Para	PopUp_Alert_Note_First PopUp_Alert_Note_Only PopUp_Alert_Important_First PopUp_Alert_Important_Only PopUp_Alert_Caution_First PopUp_Alert_Caution_Only PopUp_Alert_Warning_First PopUp_Alert_Warning_Only PopUp_Alert_Last PopUp_Alert_Mid PopUp_Para
TblTitle	TableTitle		
TblCell	CellBody, CB CellBody, CM CellBody, CT CellBody, LB CellBody, LM CellBody, LT CellBody, RB CellBody, RM CellBody, RT CellHeading, CB CellHeading, CM CellHeading, CT CellHeading, LB CellHeading, LM CellHeading, LT CellHeading, RB CellHeading, RM CellHeading, RT		
FigCaption	FigCaption		

Table 2 shows how all of the paragraph styles are grouped within the WebWorks ePublisher Pro template. Organizing paragraph styles into groups makes it possible to define HTML formatting options once, as these options are inherited by all members of the group.

Table 2 also shows the options that the sample WebWorks template applies to each paragraph style.

Table 2: WebWorks Paragraph Styles and Style Designer Options

WebWorks Paragraph Styles		**WebWorks Style Designer Options**					
WebWorks Style Groups	**WebWorks Style Names**	**Dropdown**	**Page Break Priority**	**PopUp**	**Table of Contents Level**	**Keep Paragraph Numbering**	**Generate Output**
Topics	Topic_L1	Break	1	None	1	Disabled	Enabled
	Topic_L2	Break	2	None	2	Disabled	Enabled
	Topic_L3	Break	3	None	3	Disabled	Enabled
	Topic_L4	Break	4	None	4	Disabled	Enabled
	Topic_L5	Break	5	None	5	Disabled	Enabled
DropDowns	DropDown_Closed	Start Closed	None	None	None	Disabled	Enabled
	DropDown_Open	Start Open	None	None	None	Disabled	Enabled
DropDown Paragraphs	DD_Alert_Caution_First	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Caution_Only	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Important_First	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Important_Only	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Last	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Mid	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Note_First	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Note_Only	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Warning_First	Continue	None	None	None	Disabled	Enabled
	DD_Alert_Warning_Only	Continue	None	None	None	Disabled	Enabled
	DD_Bullet_L1	Continue	None	None	None	Enabled	Enabled
	DD_Bullet_L2	Continue	None	None	None	Enabled	Enabled
	DD_Bullet_L3	Continue	None	None	None	Enabled	Enabled
	DD_Numbered_L1	Continue	None	None	None	Enabled	Enabled
	DD_Numbered_L2	Continue	None	None	None	Enabled	Enabled
	DD_Numbered_L3	Continue	None	None	None	Enabled	Enabled
	DD_Para	Continue	None	None	None	Disabled	Enabled
	DD_Para_L1	Continue	None	None	None	Disabled	Enabled
	DD_Para_L2	Continue	None	None	None	Disabled	Enabled
	DD_Para_L3	Continue	None	None	None	Disabled	Enabled
PopUps	PopUp_Exclude	Break	None	Define with no output	None	Disabled	Enabled
	PopUp_Include	Break	None	Define	None	Disabled	Enabled

Table 2: WebWorks Paragraph Styles and Style Designer Options

WebWorks Paragraph Styles		**WebWorks Style Designer Options**					
WebWorks Style Groups	**WebWorks Style Names**	**Dropdown**	**Page Break Priority**	**PopUp**	**Table of Contents Level**	**Keep Paragraph Numbering**	**Generate Output**
PopUp Paragraphs	PopUp_Alert_Caution_First	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Caution_Only	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Important_First	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Important_Only	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Last	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Mid	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Note_First	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Note_Only	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Warning_First	Break	None	Append	None	Disabled	Enabled
	PopUp_Alert_Warning_Only	Break	None	Append	None	Disabled	Enabled
	PopUp_Bullet_L1	Break	None	Append	None	Enabled	Enabled
	PopUp_Bullet_L2	Break	None	Append	None	Enabled	Enabled
	PopUp_Bullet_L3	Break	None	Append	None	Enabled	Enabled
	PopUp_Numbered_L1	Break	None	Append	None	Enabled	Enabled
	PopUp_Numbered_L2	Break	None	Append	None	Enabled	Enabled
	PopUp_Numbered_L3	Break	None	Append	None	Enabled	Enabled
	PopUp_Para	Break	None	Append	None	Disabled	Enabled
	PopUp_Para_L1	Break	None	Append	None	Disabled	Enabled
	PopUp_Para_L2	Break	None	Append	None	Disabled	Enabled
	PopUp_Para_L3	Break	None	Append	None	Disabled	Enabled
TableCells	CellBody, CB	Break	None	None	None	Disabled	Enabled
	CellBody, CM	Break	None	None	None	Disabled	Enabled
	CellBody, CT	Break	None	None	None	Disabled	Enabled
	CellBody, LB	Break	None	None	None	Disabled	Enabled
	CellBody, LM	Break	None	None	None	Disabled	Enabled
	CellBody, LT	Break	None	None	None	Disabled	Enabled
	CellBody, RB	Break	None	None	None	Disabled	Enabled
	CellBody, RM	Break	None	None	None	Disabled	Enabled
	CellBody, RT	Break	None	None	None	Disabled	Enabled
	CellHeading, CB	Break	None	None	None	Disabled	Enabled
	CellHeading, CM	Break	None	None	None	Disabled	Enabled
	CellHeading, CT	Break	None	None	None	Disabled	Enabled
	CellHeading, LB	Break	None	None	None	Disabled	Enabled
	CellHeading, LM	Break	None	None	None	Disabled	Enabled
	CellHeading, LT	Break	None	None	None	Disabled	Enabled
	CellHeading, RB	Break	None	None	None	Disabled	Enabled
	CellHeading, RM	Break	None	None	None	Disabled	Enabled
	CellHeading, RT	Break	None	None	None	Disabled	Enabled
MidTopics	Topic_Mid	Break	None	None	None	Disabled	Enabled

Table 2: WebWorks Paragraph Styles and Style Designer Options

WebWorks Paragraph Styles		**WebWorks Style Designer Options**					
WebWorks Style Groups	**WebWorks Style Names**	**Dropdown**	**Page Break Priority**	**PopUp**	**Table of Contents Level**	**Keep Paragraph Numbering**	**Generate Output**
Topic / MidTopic Paragraphs	Alert_Caution_First	Break	None	None	None	Disabled	Enabled
	Alert_Caution_Only	Break	None	None	None	Disabled	Enabled
	Alert_Important_First	Break	None	None	None	Disabled	Enabled
	Alert_Important_Only	Break	None	None	None	Disabled	Enabled
	Alert_Last	Break	None	None	None	Disabled	Enabled
	Alert_Mid	Break	None	None	None	Disabled	Enabled
	Alert_Note_First	Break	None	None	None	Disabled	Enabled
	Alert_Note_Only	Break	None	None	None	Disabled	Enabled
	Alert_Warning_First	Break	None	None	None	Disabled	Enabled
	Alert_Warning_Only	Break	None	None	None	Disabled	Enabled
	Bullet_L1	Break	None	None	None	Enabled	Enabled
	Bullet_L2	Break	None	None	None	Enabled	Enabled
	Bullet_L3	Break	None	None	None	Enabled	Enabled
	FigCaption	Break	None	None	None	Enabled	Enabled
	Numbered_L1	Break	None	None	None	Enabled	Enabled
	Numbered_L2	Break	None	None	None	Enabled	Enabled
	Numbered_L3	Break	None	None	None	Enabled	Enabled
	Para	Break	None	None	None	Disabled	Enabled
	Para_L1	Break	None	None	None	Disabled	Enabled
	Para_L2	Break	None	None	None	Disabled	Enabled
	Para_L3	Break	None	None	None	Disabled	Enabled
	TableTitle	Break	None	None	None	Disabled	Enabled
Omitted Paragraphs	WW_Omitted	Break	None	None	None	Disabled	Disabled

Building a Help System

To build a help system, an author need only do the following:

1. Create structured FrameMaker documents based on the sample page template and EDD.
2. Create a WebWorks ePublisher Pro project based on the sample WebWorks template.
3. Drag the FrameMaker files into the WebWorks project.
4. Generate the desired help system.

Creating a Customized EDD for Single-Source Publishing

This section describes the process used to create the supplied page template and EDD. The goal was to create a minimalist EDD for producing printed and on-line documentation that both gives complete control over the resulting help system and is also easy to use.

The EDD was developed in three stages. The tasks completed at each stage include the following:

- Defining the elements
- Creating rules that assign paragraph tags for WebWorks
- Creating rules that format the text for print

The enclosed CD contains a directory named EDD Development that contains a version of the sample EDD at each stage. These files are named as follows:

- Stage 1, Sample EDD.fm
- Stage 2, Sample EDD.fm
- Stage 3, Sample EDD.fm

The final step after completing the EDD is to import the EDD into a page template. The sample CD contains a directory named Structured Templates that contains the final EDD and page template. You can use the page template as is to create your own printed and on-line documents. You can also make modifications to the EDD and page template based on your own needs.

Stage One: Defining Your Elements

An EDD is a specialized FrameMaker document that defines the structure and formatting applied to structured documents. An EDD is itself a structured document. The top-level element is the ElementCatalog element.

To create a new EDD, do the following:

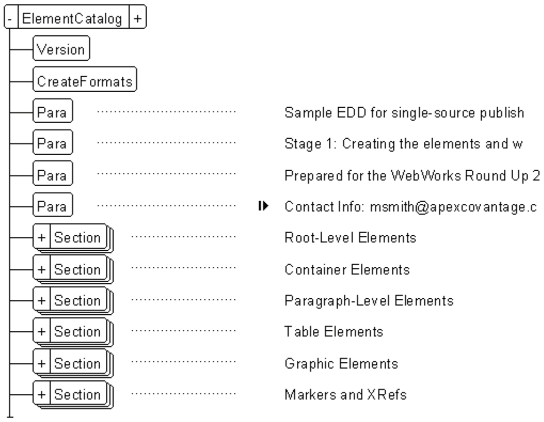
1. Select Structure Tools ▶ New EDD from the File menu.

   FrameMaker will create a new, blank EDD document.
2. Save the EDD to your disk drive.
3. Open the Elements catalog and Structure View and organize your work space so you can begin adding elements to the EDD.

Because even a minimalist EDD contains a large number of elements, we find it helpful to organize the EDD into sections. If you create a new EDD and expand the structure view, you will notice that you can add Section elements to the Element Catalog. You can then define new element types inside of each section. The sample EDD organizes elements into the following sections:

- Root-Level Elements
- Container Elements
- Paragraph-Level Elements
- Table Elements
- Graphic Elements
- Markers and X-Refs

Figure 13 shows what these sections look like in the Structure View.



ElementCatalog
Version
CreateFormats
Para Sample EDD for single-source publish
Para Stage 1: Creating the elements and w
Para Prepared for the WebWorks Round Up 2
Para Contact Info: msmith@apexcovantage.c
Section Root-Level Elements
Section Container Elements
Section Paragraph-Level Elements
Section Table Elements
Section Graphic Elements
Section Markers and XRefs

Figure 13: Adding sections to the EDD

To add sections to the EDD, do the following:

1. Expand the ElementCatalog element in the Structure View.
2. Position the cursor in the structure view to insert a new element following the CreateFormats element in the Structure View.
3. Double-click the Section element in the Elements catalog.

   FrameMaker will insert a new section in the EDD and automatically insert a Head element for the section heading.
4. Type a name for the section heading.
5. Continue adding sections to the EDD as needed.

With the EDD organized into sections, we are now ready to create the root-level elements.

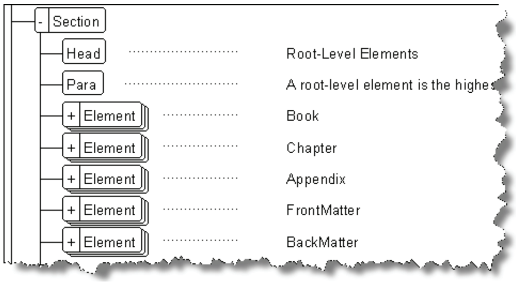
Defining Root-Level Elements

A root-level element is the highest level element in a FrameMaker document or book file. In designing the sample EDD, we decided we needed the following root-level elements:

- Book: A root-level element that represents a FrameMaker book file.

  This element is essential for any EDD used to produce books containing one or more FrameMaker documents.
- Chapter and Appendix: These elements are used to create chapters and appendices and are valid at the highest level of a FrameMaker document.
- FrontMatter: This is a general purpose element used to represent a generated or unstructured document in a FrameMaker book file. Documents that FrameMaker generates—such as a list of figures or the table of contents—are not structured documents. The FrontMatter element allows us to represent unstructured documents in a FrameMaker book file.
- BackMatter: This element serves the same purpose as the Front Matter element—it represents unstructured documents such as the index and back cover in a FrameMaker book file.

Figure 14 shows the root-level elements in the structure view.



Section
Head Root-Level Elements
Para A root-level element is the highes
Element Book
Element Chapter
Element Appendix
Element FrontMatter
Element BackMatter

Figure 14: Root-level elements in the sample EDD

▶ Creating the Book Element

To add the Book element to the EDD, do the following:

1. Expand the Root-Level Elements Section in the EDD (assuming you chose to organize the elements in your EDD by section).
2. Please the cursor in the Structure view, following the section's Head element.
3. Double-click Element in the Elements catalog.

   FrameMaker will add a new Element to the EDD. FrameMaker will automatically insert a Tag element. The Tag element is used to store the name of the element you just created.
4. Enter Book for the element tag.
5. Place the cursor in the Structure View, following your new element's Tag element.
6. Double-click Container in the Elements catalog.

   FrameMaker will automatically insert a Container element and a GeneralRule element.

   This step defines our Book element as a container element. Container elements are used to organize other elements in a structured document. Defining our Book element as a container will allow it to organize the contents of a FrameMaker book file.

Figure 15 shows what our Book element looks like in the Structure View.

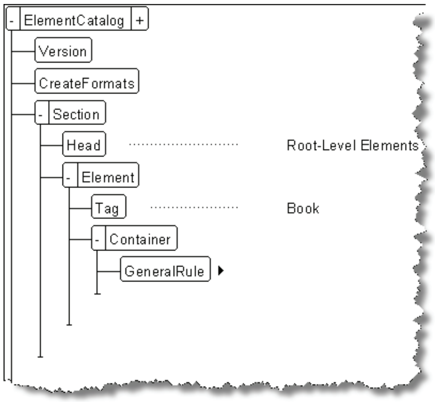


Figure 15: Defining the book element in the Structure View

At this point we have created a new container element named Book in our EDD. The next step is to write the general rule clause which specifies which elements may be included in the book element. We enter the text of the general rule clause directly in the EDD document, following the General rule heading. Figure 16 shows an example of the general rule defined for the Book element in the sample EDD.

Element (Container): Book
  General rule: FrontMatter*, Chapter+, Appendix*, BackMatter*
  Valid as the highest-level element.

Figure 16: General rule for the Book element

The text of the general rules reads as follows:

`FontMatter*, Chapter+, Appendix*, BackMatter*`

This general rule does more than simply list the elements contained in a Book element. This general rule specifies the order in which these elements must occur, and the number of times they may be repeated. Specifically, the general rule accomplishes the following:

- All of the element names in the general rule for our Book element are separated by commas. This instructs FrameMaker to require that the elements occur in the order listed: FrontMatter, Chapter, Appendix, and BackMatter.
- The asterisk following the FrontMatter, Appendix, and BackMatter elements indicates that each of these elements is optional and may occur more than once. Thus a book that only contains three chapters is structurally valid because the FrontMatter, Appendix, and BackMatter documents are optional.
- The plus sign following the Chapter element indicates that the Chapter element must occur at least once. Thus a book is structurally invalid unless it contains at least one chapter.

Once the general rule is complete, we need to indicate that the Book element is valid at the highest level of the hierarchy in the structure view. Validity at the highest level of the hierarchy is what distinguishes root-level elements from all other elements in the EDD. The sample EDD has five elements that are valid at the highest level: Book, FrontMatter, Chapter, Appendix, and BackMatter.

To indicate that the Book element is valid at the highest level do the following:

1. Expand the Book element in the Structure View.
2. Position the cursor following the Book element's GeneralRule clause in the Structure View.
3. Double-click ValidHighestLevel in the Elements catalog.

Figure 17 shows what the completed Book element looks like in the Structure View.

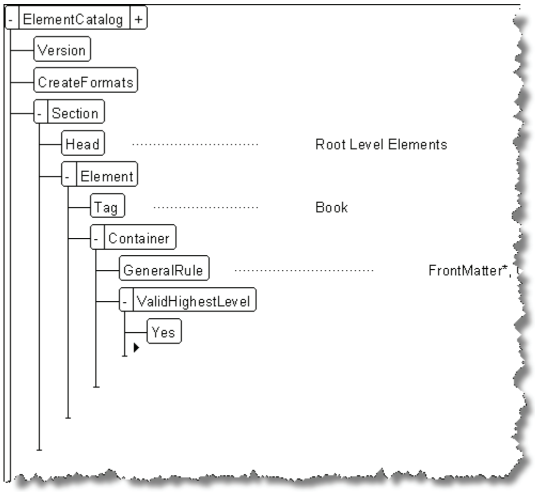


Figure 17: Completed Book element in the Structure View

Having defined the various types of elements (documents) that belong in a book file, we must now define the FrontMatter, Chapter, appendix, and BackMatter elements in the EDD.

▶ Creating the Chapter Element

The Chapter element in the sample EDD defines the elements that may occur in a FrameMaker chapter, and the order in which they occur. The process used to create the Chapter element is identical to the process used to create the Book element. The main difference between these elements is in the General Rule clause, shown in Figure 18.

Element (Container): Chapter
  General rule: ChaptNum, Title, Para, (Para | NumberedList | BulletedList | Table | Figure)*, Section+
  Valid as the highest-level element.

  Automatic insertions
    Automatically insert child: ChaptNum

Figure 18: General rule for the Chapter element

This rule clause states the following:

- A Chapter must begin with the following sequence of elements:
  - ChaptNum (Chapter Number)
  - Title (Chapter Title)
  - Para (Paragraph)

  The commas between these elements specify that the elements must occur in the order listed. The lack of any plus signs or asterisks after the element names specifies that each of these elements may only occur once.
- Once this initial sequence of elements is complete, a Chapter may contain any combination of the following element types:
  - Para
  - NumberedList
  - BulletedList
  - Table
  - Figure

  These elements are grouped by a set of parentheses and separated by vertical bars (|). The vertical bar indicates that any member of the group can occur. The parenthesis causes FrameMaker to treat the group of elements as a single element for the purpose of the occurrence indicator which follows the parenthesis—in this case an asterisk, indicating the group is optional and any member of the group may occur more than once.
- A Chapter may then contain one or more Section elements.

  The plus sign following the Section element indicates that a chapter must contain at least one section. A Section element may only be followed by another Section element.

▶ Creating the Appendix Element

The Appendix element is identical to the Chapter element. The main reason for creating distinct elements for chapters and appendices was to avoid confusion when setting the numbering properties in a FrameMaker book file. Figure 19 shows the General Rule clause of the Appendix element.

Element (Container): Appendix
Valid as the highest-level element.
General rule: AppNum, Title, Para, (Para | NumberedList | BulletedList | Table | Figure)*, Section

Automatic insertions
Automatically insert child: AppNum

Figure 19: General rule for the Appendix element

▶ Creating the FrontMatter Element

The FrontMatter element (Figure 20) is designed to represent unstructured documents in a FrameMaker book file. We typically use this element for title pages, tables of contents, and lists of figures.

The FrontMatter element has the following characteristics:

- The general rule clause contains the symbol <TEXT>.
- The FrontMatter element is valid at the highest level of a document.

This definition specifies that a document which begins with a FrontMatter element may contain any assortment of paragraphs. The document may not contain any structural elements.

Element (Container): FrontMatter
Valid as the highest-level element.
General rule: <TEXT>

Figure 20: General rule for the FrontMatter element

▶ Creating the BackMatter Element

The BackMatter element (Figure 21) is identical to the FrontMatter element. We typically use the BackMatter element to represent the index and back cover in a FrameMaker book file.

Element (Container): BackMatter
Valid as the highest-level element.
General rule: <TEXT>

Figure 21: General rule for the BackMatter element

Defining Container Elements

A container element organizes other elements. Container elements are not valid at the highest level of a document. In designing the sample EDD, we decided upon the following container elements:

- Section: A section element is followed by a heading and a series of paragraphs. Sections are the basic building blocks of chapters and become topics or mid-topics when generating a help system.
- SubSection: We created the SubSection element to implement help-specific features in a FrameMaker document. A SubSection element is followed by a heading and one or more paragraphs. The EDD formats the text of the heading in an unobtrusive manner that blends in with the surrounding text. The contents of a SubSection can become a drop-down section or pop-up window in the resulting help system.
- BulletedList
- NumberedList
- Alert: We use the Alert element to create Note, Caution, Warning, and Important statements. An attribute selects the alert type. The sample page template contains graphics on a reference page that the underlying EDD displays above and below the first and last paragraph of the alert.

Figure 22 shows the container elements in the structure view.

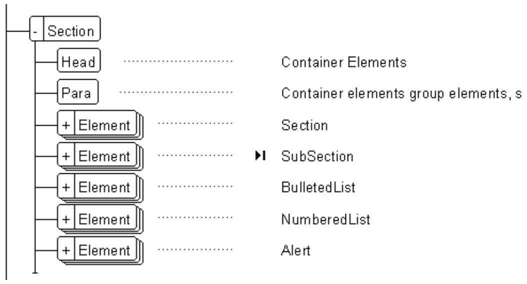


Figure 22: Container elements in the sample EDD

▶ Creating the Section Element

The Section element divides a chapter into manageable chunks of information. The contents of a section element may become a topic or mid-topic in the resulting help system. Figure 23 shows the general rule for the Section element.

Element (Container): Section
General rule: Heading, Para, (Para | Alert | NumberedList | BulletedList | Table | Figure | SubSection)*, Section*

Automatic insertions
Automatically insert child: Heading

Figure 23: General rule for the Section element

The general rule for the Section element specifies the following:

- A section must begin with a heading, followed by a paragraph.
- Once this initial sequence of elements is complete, a section may contain any combination of paragraphs, alerts, numbered lists, bulleted lists, tables, figures, and sub-sections.
- A section may then contain one or more section elements. The ability to nest sections inside of sections creates different levels of headings, similar to the Heading 1, Heading 2, Heading 3 paragraph styles typically used in an unstructured document.

▶ Creating the SubSection Element

The SubSection element will enable us to create drop-down sections and pop-up windows in our FrameMaker source files. We will add the logic that implements this behavior in the next phase. For now, we need to define the various types of elements that may be included in a SubSection. Figure 24 shows the general rule for the SubSection element.

Element (Container): SubSection
General rule: Heading, Para, (Para | Alert | NumberedList | BulletedList)*

Automatic insertions
Automatically insert child: Heading

Figure 24: General rule for the SubSection element

The general rule for the SubSection element specifies the following:

- A sub-section must begin with a heading, followed by an initial paragraph.
- Once this initial sequence of elements is complete, a sub-section may contain any combination of paragraphs, alerts, numbered lists, and bulleted lists.

▶ Creating the BulletedList Element

Figure 25 shows the general rule for the BulletedList element.

Element (Container): BulletedList
General rule: ListMember, (ListMember | Para | Table | Figure)*

Automatic insertions
Automatically insert child: ListMember

Figure 25: General rule for the BulletedList element

The general rule for the BulletedList element specifies the following:

- A bulleted list must begin with a ListMember.

  We will define the ListMember to automatically insert bulleted or numbered list items, depending on the list type.
- Subsequent elements in a bulleted list may include additional list members, paragraphs, tables, and figures, in any order.

Permitting tables and figures to occur in a bulleted list was an editorial decision. If desired, you could easily modify the EDD to exclude tables and figures from bulleted lists. Another editorial decision was to prevent authors from creating nested, bulleted lists.

▶ Creating the NumberedList Element

Figure 26 shows the general rule for the BulletedList element.

Element (Container): NumberedList
General rule: ListMember, (ListMember | Para | Table | Figure | NumberedList | BulletedList)*

Automatic insertions
Automatically insert child: ListMember

Figure 26: General rule for the NumberedList element

The general rule for the NumberedList element specifies the following:

- A numbered list must begin with a list member.
- A numbered list may then contain any assortment of list members, paragraphs, tables, figures, numbered lists, or bulleted lists.

Unlike the BulletedList element, we have chosen to allow authors to create nested, numbered lists. The intelligence for applying the nested numbering scheme will reside within the ListMember element. We have also allowed numbered lists to contain nested, bulleted lists. These are editorial decisions that one can make when designing an EDD. This is also another area where you may want to modify the sample EDD to comply with your own editorial standards.

▶ Creating the Alert Element

The Alert element will allow us to create Note, Caution, Warning, and Important statements. At this stage, we simply need to create the Alert element and write the general rule, shown in Figure 27.

Element (Container): Alert
General rule: Para+

Automatic insertions
Automatically insert child: Para

Figure 27: General rule for the Alert element

The general rule for the Alert element specifies the following:

- The Alert must contain one or more paragraphs.

In the next stage we will add the attributes that allows the author to format the alert as a Note, Caution, Warning, or Important statement. The intelligence for formatting the text will reside within the Para element.

Defining Paragraph-Level Elements

Paragraph-level elements store text. Paragraph-level elements may also contain in-line elements such as markers and cross references.

We strove to keep the number of paragraph-level elements to an absolute minimum in order to create a single-source authoring environment that is easy for authors to learn and use. The paragraph-level elements in the sample EDD include the following:

- ChaptNum: This element assigns the chapter number to a FrameMaker document. The sample template uses the numbering features of the FrameMaker book file to control chapter, appendix, and page numbering.
- AppNum: The AppNum element assigns the appendix number to a FrameMaker document. The actual numbering properties of an appendix depend on the numbering options specified in the FrameMaker book file.
- Title: This element is used to create the title of a chapter or appendix.
- Heading: This element is used for all headings inside Section and SubSection elements.
- ListMember: This element is used to create members of bulleted and numbered lists.
- Para: The Para element creates a paragraph. The Para element is the most complicated element in the EDD, as it can occur inside of a chapter, appendix, section, sub-section, numbered list, bulleted list, or table cell.

Figure 28 shows the paragraph-level elements in the structure view.

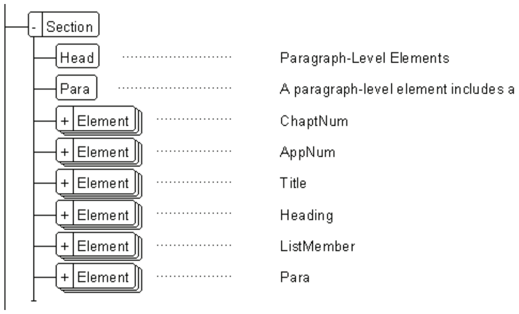


Figure 28: Paragraph-level elements in the EDD

▶ Creating the ChaptNum Element

The ChaptNum element assigns the chapter number. Figure 29 shows the general rule for the ChaptNum element.

Element (Container): ChaptNum
General rule: <TEXT>

Figure 29: General rule for the ChaptNum element

The general rule for the ChaptNum element specifies the following:

- The ChaptNum element contains text.

In Stage 3 we will add the logic that enables FrameMaker to number chapters automatically and format the text.

▶ Creating the AppNum Element

The AppNum element assigns the appendix number. Figure 30 shows the general rule for the AppNum element.

Element (Container): AppNum
General rule: <TEXT>

Figure 30: General rule for the AppNum element

The general rule and operation of the AppNum element is identical to the ChaptNum element.

▶ Creating the Title Element

The Title element is used to create the title of a chapter or appendix. The title will also become a first-level topic in the resulting help system. Figure 31 shows the general rule for the Title element.

Element (Container): Title
General rule: PageStyle?, (<TEXT> | Index)*, TopicAlias?

Attribute list
1. Name: ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Figure 31: General rule for the Title element

The general rule for the Title element specifies the following:

- The title may begin with an optional PageStyle element. The PageStyle element inserts a PageStyle marker at the beginning of the title, allowing WebWorks ePublisher Pro to apply a custom page style to the resulting topic. The question mark following the PageStyle element indicates that this element is optional.
- The title may then contain an assortment of text and index markers.
- The title may end with an optional TopicAlias element. The TopicAlias element inserts a TopicAlias marker used to create context-sensitive help systems.

▶ Creating the Heading Element

The Heading element contains the text of headings used in Section and SubSection elements. The text of a heading element may also become the text of a topic, mid-topic, drop-down, or pop-up heading in the resulting help system. Figure 32 shows the general rule for the Heading element.

Element (Container): Heading
General rule: (<TEXT> | Index)*, TopicAlias?

Attribute list
1. **Name:** ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Figure 32: General rule for the Heading element

The general rule for the Heading element specifies the following.

- The heading may contain any combination of text and index markers.
- The text of the heading may be followed by an optional TopicAlias marker, which is used to create context-sensitive help systems.

▶ Creating the ListMember Element

The ListMember element contains the text displayed in numbered and bulleted lists. Figure 33 shows the general rule for the ListMember element.

Element (Container): ListMember
General rule: (<TEXT> | XRef | Index | FigFrame)*

Figure 33: General rule for the ListMember element

The general rule for the ListMember element specifies the following:

- A list member may contain any combination of text, cross references, index markers, and anchored frames.

The FigFrame element is intended to allow authors to insert small graphics in-line with the text.

▶ Creating the Para Element

The Para element is the most commonly used element in the sample EDD and page template. At this stage we simply need to write the general rule clause. In the second and third stages we will write the rules that automatically apply paragraph tags and format the text in each context that the Para element is used.

Figure 34 shows the general rule for the Heading element.

Element (Container): Para
General rule: (<TEXT> | XRef | Index | FigFrame)*

Figure 34: General rule for the Para element

The general rule for the Para element specifies the following:

- A paragraph may contain any combination of text, cross references, index markers, and anchored frames.

The FigFrame element is intended to allow authors to insert small graphics in-line with the text.

Defining Table Elements

FrameMaker uses an assortment of element types to represent tables in structured documents. (So far we have only worked with container elements.) You need to create instances of all the table-related elements when creating an EDD. Table 3 summarizes all of Frame Maker's table-specific element types.

The table also lists the names of these elements as defined in the sample EDD and page template.

Table 3: Table-specific elements in the EDD

Element Type	Element Name	Function
Table	Table	This element type represents a table in a FrameMaker document. This is the top-level element of a table, as represented in the Structure View.
TableTitle	TblTitle	This element type represents the title of a table. The formatting applied to the table determines whether the title is displayed above or below the table. The title can also be omitted from a table.
TableHeading	TblHeading	This element represents the heading of the table, which is optional. The heading may consist of one or more rows. Heading rows repeat when a table spans multiple pages.
TableBody	TblBody	This element represents the body of a table. The body of a table consists of one or more rows. A table is required to contain at least one body row.
TableFooting	TblFooting	This element represents the footing of a table, which is optional. The footing may consist of one or more rows. Footer rows repeat when a table spans multiple pages.
TableRow	HeadingRow	This element represents a row in a table. Rows are contained within TableHeading, TableBody, and TableFooting type elements. We chose to create three TableRow-type elements in the EDD, for use inside the heading, body, and footing sections of a table. Doing so simplified the logic required to format the text in the EDD.
	BodyRow	
	FootingRow	
TableCell	TblCell	This element represents a cell in a table. A cell may contain text or one or more elements.

Note that the EDD only defines the structure of tables in a FrameMaker document. The way tables are formatted is still determined by the table format defined in the page template using the Table Designer. This is different than the way FrameMaker treats paragraphs in a structured document, where the formatting can be controlled entirely from the EDD.

Figure 35 shows the table-specific elements defined in the sample EDD, as shown in the Structure View.

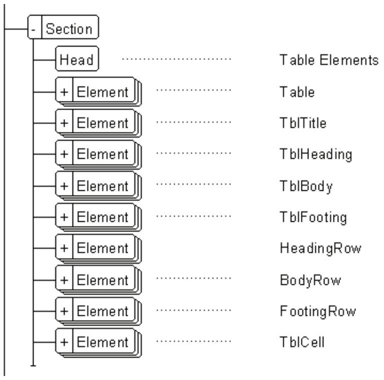


Figure 35: Table-specific elements in the EDD

▶ Creating the Table Element

The Table element is the highest level element of a table in a structured FrameMaker document. Figure 36 shows the general rule for the Table element.

Element (Table): Table
General rule: TblTitle?, TblHeading?, TblBody, TblFooting?

Figure 36: General rule for the Table element

The general rule for the Table element specifies the following:

- A table consists of the following elements: table title, table heading, table body, and table footing. Of these elements, only the table body is required. The other elements are optional. The elements must occur in the order listed.

Note that you do not have to insert these elements when you create a table. All of the table elements are inserted automatically when you insert a table. We simply need to define the elements of the table in the EDD in order for tables to function properly in a structured FrameMaker document.

▶ Creating the TblTitle Element

The TblTitle element represents a table title. Figure 37 shows the general rule for the TblTitle element.

Element (Table Title): TblTitle
General rule: (<TEXT> | Index)*

Attribute list
1. **Name:** ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Figure 37: General rule for the TblTitle element

The general rule for the TblTitle element specifies the following:

- A table title may contain any combination of text and index markers. A table title may not contain any paragraph-level or in-line elements.

▶ Creating the TblHeading Element

The TblHeading element represents a table heading in a structured FrameMaker document. The table heading element contains the heading rows. These rows repeat when a table spans multiple pages. Figure 38 shows the general rule for the TblHeading element.

Element (Table Heading): TblHeading
General rule: HeadingRow+

Figure 38: General rule for the TblHeading element

The general rule for the TblHeading element specifies the following:

- A table heading may contain one or more optional heading rows.

▶ Creating the TblBody element

The TblBody element represents the body of a table in a structured FrameMaker document. The body contains one or more table rows. Figure 39 shows the general rule for the TblBody element.

Element (Table Body): TblBody
General rule: BodyRow+

Figure 39: General rule for the TblBody element

The general rule for the TblBody element specifies the following:

- The body of a table must contain one or more body rows.

▶ Creating the TblFooting Element

The TblFooting element represents the footing of a table in a structured FrameMaker document. A table footing contains one or more optional footing rows. Footing rows repeat when a table spans more than one page. Figure 40 shows the general rule for the TblFooting element.

Element (Table Footing): TblFooting
General rule: FootingRow+

Figure 40: General rule for the TblFooting element

The general rule for the TblFooting element specifies the following:

- A table footing may contain one or more optional footing rows.

▶ Creating the HeadingRow, BodyRow, and FootingRow Elements

The HeadingRow, BodyRow, and FootingRow elements represent a row in a table heading, body, and footing respectively. We chose to create separate row elements for use in the heading, body, and footing sections of a table to simplify the logic necessary to format the text. Figure 41 shows the general rule for each element.

Element (Table Row): HeadingRow
General rule: TblCell+

Element (Table Row): BodyRow
General rule: TblCell+

Element (Table Row): FootingRow
General rule: TblCell+

Figure 41: General rule for the HeadingRow, BodyRow, and FootingRow elements

The general rule for each of these row elements is identical and specifies the following:

- A row consists of one or more table cells. A row must contain at least one cell.

▶ Creating the TblCell Element

The TblCell element represents a cell in a table. A cell may contain text or one or more paragraphs. Figure 42 shows the general rule for the TblCell element.

Element (Table Cell): TblCell
General rule: (<TEXT> | FigFrame | XRef | Index)* | (Para | NumberedList | BulletedList | Figure)*

Figure 42: General rule for the TblCell element

The general rule for the TblCell element contains two groups of elements enclosed in parenthesis and separated by a vertical bar (|). The general rule specifies the following:

- The first group of elements (<TEXT> | FigFrame | XRef | Index) specifies that a table cell may contain any combination of text, graphics, cross references, and index markers.
- The second group of elements (Para | NumberedList | BulletedList | Figure) specifies that a table cell may contain any assortment of paragraphs, numbered lists, bulleted lists, and figures. There is no need to include cross references and index markers in this second grouping because all of the elements in this second grouping may themselves contain cross references and index markers.

The author can decide whether to begin typing directly in a table cell (in which case the first grouping applies) or insert elements into the table cell from the Elements catalog (in which case the second grouping applies). If you want to include more than one paragraph in a table cell you need to insert elements from the Elements catalog.

Defining Graphic Elements

The sample EDD and page template provides three elements for working with figures and figure captions. The sample EDD supports in-line graphics and figures. Figure 43 shows the graphic elements defined in the EDD, as shown in the Structure View.

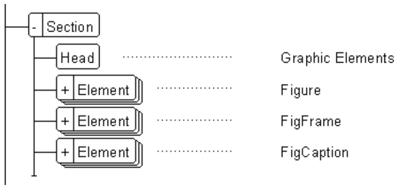


Figure 43: Graphic-specific elements in the EDD

▶ Creating the Figure Element

The Figure element is used to insert an anchored frame and an optional figure caption. The Figure element is not intended for creating in-line graphics. To insert in-line graphics, use the FigFrame element. Figure 44 shows the general rule for the Figure element.

Element (Container): Figure
General rule: FigFrame, FigCaption?

Automatic insertions
Automatically insert child: FigFrame

Figure 44: General rule for the Figure element

The general rule for the Figure element specifies the following:

- A figure consists of a an anchored frame (FigFrame) and an optional figure caption.

▶ Creating the FigFrame Element

The FigFrame element inserts an anchored frame in a FrameMaker document. Figure 45 shows the definition of the FigFrame element.

Element (Graphic): FigFrame

Initial graphic element format

1. In all contexts.
Insert anchored frame.

Figure 45: General rule for the FigFrame element

The definition of the FigFrame element specifies that FrameMaker inserts an anchored frame.

▶ Creating the FigCaption Element

Figure 46 shows the general rule for the FigCaption element.

Element (Container): FigCaption
General rule: (<TEXT> | Index)*

Attribute list
1. Name: ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Figure 46: General rule for the FigCaption element

The general rule for the FigCaption element specifies the following:

- A figure caption may contain any combination of text and index markers.

Defining Markers and X-Refs

The sample EDD and page template includes specialized elements for inserting markers and cross references. Figure 47 shows the marker and cross reference elements defined in the EDD, as shown in the Structure View.

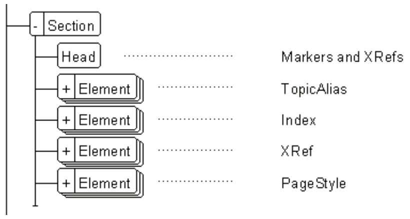


Figure 47: Marker and cross reference elements in the EDD

▶ Creating the TopicAlias Element

The TopicAlias element is used to insert TopicAlias markers that become context-sensitive links in the resulting help system. Figure 48 shows the definition of the TopicAlias element.

Element (Marker): TopicAlias

Initial marker type

1. **In all contexts.**
   Use marker type: TopicAlias

Figure 48: Definition of the TopicAlias element

The definition of the TopicAlias element specifies the following:

- The TopicAlias element inserts a marker in the text named TopicAlias.

▶ Creating the Index Element

The Index element is used to insert Index markers in the text. Figure 49 shows the definition of the Index element.

Element (Marker): Index

Initial marker type

1. **In all contexts.**
   Use marker type: Index

Figure 49: Definition of the Index element in the EDD

The definition of the Index element specifies the following:

- The Index element inserts a marker in the text named Index.

▶ Creating the XRef Element

The XRef element represents an element-based cross reference in a FrameMaker document. The actual format of the cross reference is defined in the page template, not the EDD. Cross references in structured documents are based on a unique identification number that FrameMaker uses to align the cross reference with its target. The sample EDD is designed to allow element-based cross references between the XRef cross reference element and the following targets: chapter and appendix titles, headings, figure captions, and table titles. Figure 50 shows the definition of the XRef element.

Element (CrossReference): XRef

Attribute list

1. **Name:** XRefID **ID Reference** **Required**
   Control flags: Read-only

Figure 50: Definition of the XRef element

The definition of the XRef element specifies the following:

- The XRef element creates a cross reference to a target element based on its XRefID attribute. The value of this attribute is assigned automatically by FrameMaker and is invisible to the user.

Enabling and using element-based cross references is necessary because if we were to create paragraph-based cross references using the sample page template and EDD, we would have to create cross references based on the paragraph tags the EDD assigns for WebWorks benefit. Thus in order to create a cross reference to a heading, we would have to know whether the heading is formatted as a topic, mid-topic, pop-up, or drop-down in the resulting help system. Using element-based cross references eliminates this problem.

In terms the interaction between FrameMaker and the help system, WebWorks processes cross references based on the name of the cross reference format, as defined in the page template. The underlying EDD does not specify the building blocks used to form the text of the cross reference.

Stage 2:
Creating Rules that Assign Paragraph Tags for WebWorks

Having defined all of the elements in the EDD and specified the general rule clauses, we are now ready to add Attributes and Text Format Rules to the EDD. We will add attributes to Section and SubSection container elements that will allow authors to choose between topic / mid-topic breaks and create pop-ups and drop-down sections. We will then add Text Format Rules to paragraph-level elements. These Text Format Rules will allow FrameMaker to automatically tag paragraphs with names that align with our WebWorks ePublisher Pro template. In the final stage we will add Text Format Rules that format paragraph-level elements for print.

Creating Attributes to Control Help-Specific Features

An EDD uses Text Format Rules to format paragraph-level elements based on context. For example, we can design the EDD to format the content of a ListMember element with a bullet character when it appears within a BulletedList and a number if it appears within a NumberedList. An EDD can make very complex formatting decisions because it can detect the context in which an element is used a document. The EDD is able to distinguish between a ListMember that appears in a BulletedList from a ListMember that appears in a Numbered List.

In addition, the EDD is able to make formatting decisions based on the value of attributes. This ability to make formatting decisions based on attribute values becomes especially powerful when combined with contextual awareness. In this way we are able to automatically assign paragraph tags to every paragraph in a SubSection based on whether the author wants to display the text as a pop-up or drop-down in the resulting help system.

The first step is to add attributes to our Section, SubSection, and Alert container elements that we can then detect when we write the underlying logic for our paragraph-level elements.

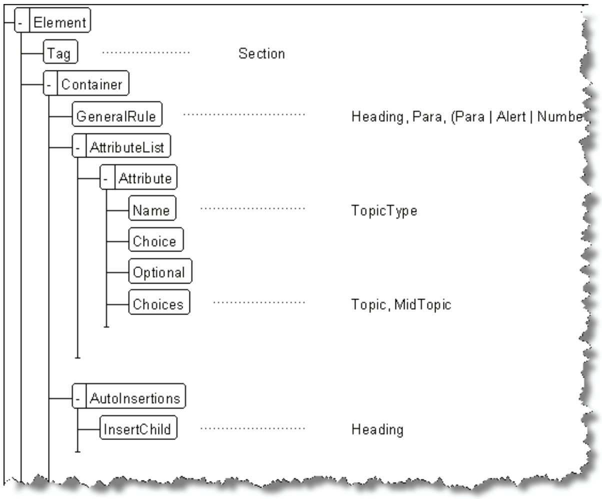
▶ Controlling Topic / Mid-Topic Breaks

The sample EDD and page template controls topic / mid-topic breaks at the Section level. Recall that nesting Sections in our structured document creates the same cascade of headings (Heading 1, Heading 2, etc.) found in an unstructured document.

In order to give the author control over whether a section gets formatted as a topic or mid-topic in the resulting help system we need to add an attribute named TopicType to the EDD. The TopicType attribute has two possible values: Topic and MidTopic.

We begin by adding an AttributeList element to the Section element, following the GeneralRule clause. An AttributeList may contain one or more attributes. We then add the TopicType attribute to the list and configure the attribute to present the author with list of choices when a new section is added to a document. The choices for TopicType include Topic and MidTopic.

Figure 51 shows what the Section element looks like in the Structure View after adding the AttributeList and the TopicType Attribute. Figure 51 also shows what the Section element definition looks like in the EDD itself. We have highlighted what we added to the EDD in blue. The element definition and general rule clause remain unchanged.



Element (Container): Section
General rule: Heading, Para, (Para | Alert | NumberedList | BulletedList | Table | Figure | SubSection)*, Section*

Attribute list
1. Name: TopicType **Choice** **Optional**
Choices: Topic, MidTopic

Automatic insertions
Automatically insert child: Heading

Figure 51: Adding attributes to the Section element

The new AttributeList specifies the following:

- The Section element contains one attribute named TopicType.
- The TopicType attribute is configured as a Choice attribute—meaning FrameMaker displays a list of possible attribute values in a drop-down menu when an author inserts a new section in a document.
- Two choices are available for TopicType: Topic and MidTopic.
- The TopicType attribute is optional, meaning FrameMaker will not force the author to make a decision when he or she creates a new topic. When we write the underlying logic, we will format section headings as mid-topics if the author does not make a selection for TopicType.

▶ Controlling Pop-Ups and Drop-Downs

Like the Section element, the SubSection element contains a choice attribute that allows the author to select help-specific features from within the FrameMaker source files. Figure 52 shows that the SubSection element definition looks like in the EDD. We highlighted the specification for the attribute list in blue. The element definition and general rule remain unchanged.

Element (Container): SubSection
General rule: Heading, Para, (Para | Alert | NumberedList | BulletedList)*

Attribute list
1. Name: TopicType **Choice** **Optional**
Choices: DropDown: Start Closed, DropDown: Start Open, PopUp: Include Text, PopUp: Exclude Text

Automatic insertions
Automatically insert child: Heading

Figure 52: Adding attributes to the SubSection element

The new AttributeList specifies the following:

- The SubSection element contains one attribute named TopicType.
- The TopicType attribute is configured as a Choice attribute. FrameMaker will display a drop-down menu when an author inserts a SubSection in a document.
- The following choices are available for TopicType:
  - DropDown: Start Closed
  - DropDown: Start Open
  - PopUp: Include Text
  - PopUp: Exclude Text.
- The TopicType attribute is optional.

▶ Selecting Note, Caution, Warning, and Important Statements for Alerts

The Alert element contains an attribute named AlertType that allows an author to select the alert type. Options include Note, Caution, Warning, and Important. Figure 53 shows that the SubSection element definition looks like in the EDD.

Element (Container): Alert
General rule: Para+

Attribute list
1. Name: AlertType **Choice** **Required**
Choices: Note, Caution, Warning, Important

Automatic insertions
Automatically insert child: Para

Figure 53: Adding attributes to the Alert element

The AlertType attribute specifies the following:

- An author can select from the following alert types: Note, Caution, Warning, and Important.
- The AlertType is required. An author must select an alert type before FrameMaker will insert an alert in the document.

Assigning Paragraph Tags that Align with the WebWorks Template

Having added attributes to the Section and SubSection container elements, we are ready to write the text formatting rules that will assign paragraph tags based on context and the author's attribute selections.

We begin by inserting a TextFormatRule element following an element's attribute list (if present) or general rule clause. Figure 54 shows an example of what a TextFormatRule looks like in the Structure View.

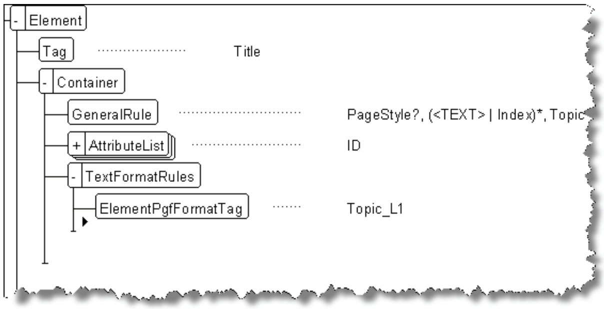

Element
Tag .... Title
Container
GeneralRule .... PageStyle?, (<TEXT> | Index)*, Topic
AttributeList .... ID
TextFormatRules
ElementPgfFormatTag .... Topic_L1

Figure 54: Text format rule for the Title element

A text format rule element is followed by one of the following types of rules:

- AllContextsRule—An all contexts rule applies formatting to an element regardless of the context in which the element is used. We use all context rules to apply formatting—such as the font family—that does not change from one context to another.
- ContextRule—We typically use context rules apply formatting to an element based on the name of an element at a higher level in the hierarchy. Context rules can also apply formatting changes based on the value of one or more attributes.
- LevelRule—We use level rules to detect an element's nesting level. We use level rules in the sample EDD and page template to format different levels of headings and to properly indenture members of bulleted and numbered lists.

Context and level rules may be combined and nested to create complex sets of conditions for assigning paragraph tags and formatting text.

Following are some important points to keep in mind when writing text format rules for WebWorks ePublisher Pro workflows:

- If you are assigning paragraph tags for the benefit of WebWorks, always assign the paragraph tag before including additional logic to format the text. This principle is critical for success. If you attempt to format the text and then apply the paragraph tag, FrameMaker's default paragraph style definition will take precedence over your formatting changes.
- When an element contains two or more context or level rules at the same nesting level, FrameMaker does the following:
  - FrameMaker applies any formatting rules that evaluate as true in the first rule.
  - FrameMaker then applies any formatting rules that evaluate as true in the second rule.
  - Framemaker then applies any formatting rules that evaluate as true in the third rule.
  - Etc.

  For this reason, we recommend structuring the context and level rules in an EDD based on the following strategy:
  - The very first rule should always be a context rule that applies paragraph formats for every context an element can be used in a document.
  - The second rule should specify paragraph indents and tabs for every context an element can be used in a document.
  - The third rule should apply spacing above and below the paragraph for every context an element can be used in a document.
  - The fourth rule should specify font family, size, and weight for every context an element can be used in a document.
  - The fifth rule should specify any advanced paragraph properties for every context an element can be used in a document.

  Figure 55 shows how we applied this principle when writing the context rules for the list member element.

▶ Tagging the Chapter and Appendix Number

We use the chapter number and appendix number elements (ChaptNum and AppNum) to number chapters and appendices in FrameMaker books. Because there is no counterpart in an on-line help system, we apply paragraph tags to the corresponding paragraphs that cause WebWorks to omit chapter and appendix numbers from the help system. The paragraph style in the sample WebWorks template that omits paragraphs from the help system is named WW_Omitted.

Figure 56 shows the text format rules that apply the WW_Omitted paragraph tag to the chapter and appendix numbers.

Element (Container): ChaptNum
General rule: <TEXT>

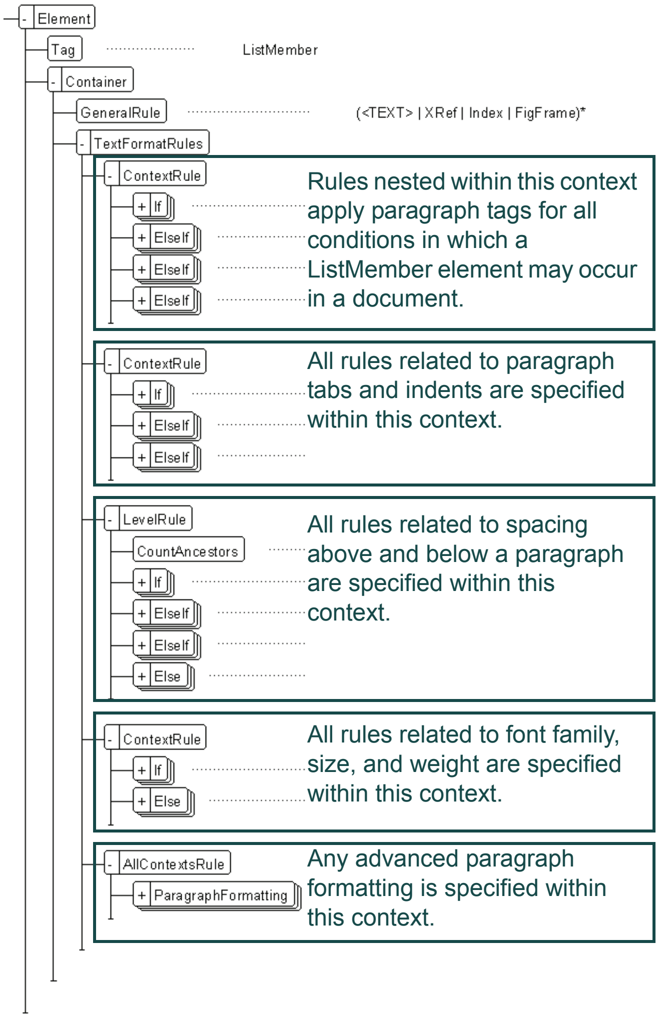
Text format rules
Element paragraph format: WW_Omitted

Element (Container): AppNum
General rule: <TEXT>

Text format rules
Element paragraph format: WW_Omitted

Figure 56: Assigning a paragraph tags to the ChaptNum and AppNum elements

The ChaptNum and AppNum elements are only used in one context—at the beginning of a Chapter or Appendix.



Rules nested within this context apply paragraph tags for all conditions in which a ListMember element may occur in a document.

All rules related to paragraph tabs and indents are specified within this context.

All rules related to spacing above and below a paragraph are specified within this context.

All rules related to font family, size, and weight are specified within this context.

Any advanced paragraph formatting is specified within this context.

Figure 55: Developing context formatting rules for paragraph tags, tabs and indents, spacing, font settings, and advanced properties

Consequently we can simply assign the WW_Omitted paragraph tag using the ElementPgfFormatTag element immediately following the general rule clause.

▶ Tagging the Title Element

Likewise, the Title element is only used within one context in a FrameMaker document—as the title of a Chapter or Appendix. Figure 57 shows the text format rule that applies a paragraph tag to the Title element.

Element (Container): Title
General rule: PageStyle?, (<TEXT> | Index)*, TopicAlias?

Attribute list
1. **Name:** ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Text format rules
Element paragraph format: Topic_L1

Figure 57: Assigning a paragraph tag to the Title element

Chapter and Appendix titles always become first-level topics in the enclosed WebWorks template. Consequently we can assign the Topic_L1 paragraph tag to the title using an ElementPgfFormatTag element immediately following the general rule clause.

▶ Tagging the Heading Element

The sample EDD and page template tags the paragraph associated with a heading element with one of nine paragraph styles. The sample EDD uses one context rule to make all of these paragraph assignments. Nested within the context rule is a level rule that assigns paragraph tags for new topics based on a section's nesting level in the document structure.

Figure 58 shows the text format rules that apply the nine possible paragraph tags to paragraphs associated with the Heading element.

Element (Container): Heading
General rule: (<TEXT> | Index)*, TopicAlias?

Attribute list
1. **Name:** ID **Unique ID** **Required**
Control flags: Read-only, Hidden

Text format rules

1. **If context is:** Section[TopicType = "Topic"]
    1.1. Count ancestors named: Section
        If level is: 1
            Use paragraph format: Topic_L2
        Else, if level is: 2
            Use paragraph format: Topic_L3
        Else, if level is: 3
            Use paragraph format: Topic_L4
        Else, if level is: 4
            Use paragraph format: Topic_L5
        Else
            Use paragraph format: Topic_L5
    Else, if context is: Section[TopicType = “MidTopic”]
        Use paragraph format: Topic_Mid
    Else, if context is: Section[TopicType = “”]
        Use paragraph format: Topic_Mid
    Else, if context is: SubSection[TopicType = “DropDown: Start Closed”]
        Use paragraph format: DropDown_Closed
    Else, if context is: SubSection[TopicType = “DropDown: Start Open”]
        Use paragraph format: DropDown_Open
    Else, if context is: SubSection[TopicType = “PopUp: Include Text”]
        Use paragraph format: PopUp_Include
    Else, if context is: SubSection[TopicType = “PopUp: Exclude Text”]
        Use paragraph format: PopUp_Exclude
    Else, if context is: SubSection[TopicType = “”]
        Use paragraph format: Topic_Mid

Figure 58: Assigning paragraph tags to the Heading element

Some notable aspects of the context rule that applies the paragraph tags include the following:

- The outer context rule consists of one If clause and seven ElseIf clauses that describe all of the contexts in which a Heading element can occur within a document based on the sample EDD.
- These eight outer clauses examine the name of the parent element that contains the Heading (Section element or SubSection element) as well as the value of the TopicType attribute selected by the author.
- When the author selects Topic for TopicType, the EDD evaluates how deeply the parent section is nested in the document structure and makes paragraph tag assignments accordingly. Depending on how deeply a section is nested, the EDD assigns paragraph styles Topic_L2, Topic_L3, Topic_L4, and Topic_L5.

▶ Tagging the ListMember Element

Figure 59 shows the context rule that applies paragraph tags to the ListMember element. This one context rule assigns one of 27 different paragraph tags to paragraphs associated with ListMember elements. The reason why we have so many paragraph tags for list members is because of the following considerations:

- A list member may occur in a numbered list.
- A list member may occur in a bulleted list.
- Numbered and bulleted lists can be nested up to three layers deep.
- All list combinations can occur in help topics, mid-topics, drop-down sections, and pop-up windows.

Although the underlying context rule in the EDD is complex, this complexity is completely hidden from the author. The author can create topics, mid-topics, drop-downs, and pop-ups and proceed to add content without thinking about the paragraph tags necessary to create the help system.

The key to writing complex rules such is this is to

- Keep careful track of all the contexts in which an element can be used in a document.
- Begin by specifying the most specific case, working towards the most general case. This is essential because FrameMaker stops executing a context rule once a statement evaluates as true. Said another way, FrameMaker will never evaluate to the specific case if it encounters the general case first.

Finally, we recommend developing EDDs that simplify the workflow for writers as much as possible. This principle has the following implications when developing an EDD:

- Resist the temptation to add additional elements to the EDD to simplify the underlying logic. In the case of the sample EDD, the context rule would have been easier to write if we required the author to use different elements inside numbered and bulleted lists. Likewise the Alert element would have been easier to develop had we created separate elements for Note, Caution, Warning, and Important statements.
- If you resist the temptation to add additional elements, some aspects of EDD development actually become more simple. Having fewer elements means keeping track of fewer contexts in which those elements can occur.

```
Element (Container): ListMember
    General rule:   (<TEXT> | XRef | Index | FigFrame)*

    Text format rules

        1.  If context is: * < (SubSection[TopicType = “DropDown: Start Closed”] |
            SubSection[TopicType = “DropDown: Start Open”])

                1.1. If context is: BulletedList
                        1.1.1.Count ancestors named:  BulletedList, NumberedList
                            If level is: 1
                                Use paragraph format: DD_Bullet_L1
                            Else, if level is: 2
                                Use paragraph format: DD_Bullet_L2
                            Else, if level is: 3
                                Use paragraph format: DD_Bullet_L3
                            Else
                                Use paragraph format: DD_Bullet_L3
                    Else, if context is: NumberedList
                        1.1.2.Count ancestors named:  NumberedList
                            If level is: 1
                                Use paragraph format: DD_Numbered_L1
                            Else, if level is: 2
                                Use paragraph format: DD_Numbered_L2
                            Else, if level is: 3
                                Use paragraph format: DD_Numbered_L3
                            Else
                                Use paragraph format: DD_Numbered_L3
            Else, if context is: * < (SubSection[TopicType = “PopUp: Include Text”] |
            SubSection[TopicType = “PopUp: Exclude Text”])

                1.2. If context is: BulletedList
                        1.2.1.Count ancestors named:  BulletedList, NumberedList
                            If level is: 1
                                Use paragraph format: PopUp_Bullet_L1
                            Else, if level is: 2
                                Use paragraph format: PopUp_Bullet_L2
                            Else, if level is: 3
                                Use paragraph format: PopUp_Bullet_L3
                            Else
                                Use paragraph format: PopUp_Bullet_L3
                    Else, if context is: NumberedList
                        1.2.2.Count ancestors named:  NumberedList
                            If level is: 1
                                Use paragraph format: PopUp_Numbered_L1
                            Else, if level is: 2
                                Use paragraph format: PopUp_Numbered_L2
                            Else, if level is: 3
                                Use paragraph format: PopUp_Numbered_L3
                            Else
                                Use paragraph format: PopUp_Numbered_L3
            Else, if context is: BulletedList
                1.3. Count ancestors named:    BulletedList, NumberedList
                    If level is: 1
                        Use paragraph format: Bullet_L1
                    Else, if level is: 2
                        Use paragraph format: Bullet_L2
                    Else, if level is: 3
                        Use paragraph format: Bullet_L3
                    Else
                        Use paragraph format: Bullet_L3
            Else, if context is: NumberedList
                1.4. Count ancestors named:    NumberedList
                    If level is: 1
                        Use paragraph format: Numbered_L1
                    Else, if level is: 2
                        Use paragraph format: Numbered_L2
                    Else, if level is: 3
                        Use paragraph format: Numbered_L3
                    Else
                        Use paragraph format: Numbered_L3
```

Figure 59: Assigning paragraph tags to the ListMember element

▶ Tagging the Para Element.

Figure 60 shows the context rule used to assign one of 36 different paragraph tags to paragraphs associated with the Para element. The Para element is the most complicated element in the EDD. The logic behind the Para element also hides a great deal of complexity from the author.

Some notable aspects of this context rule include the following:

- At the outermost level, the context rule for the Para element works methodically from the most specific case to the most general case.

```
Element (Container): Para
    General rule:   (<TEXT> | XRef | Index | FigFrame)*

    Text format rules

        1.  If context is: * < (SubSection[TopicType = "DropDown: Start Closed"] |
            SubSection[TopicType = "DropDown: Start Open"])

                1.1. If context is: Alert[AlertType = "Important"]

                        1.1.1.If context is: {only}
                                Use paragraph format: DD_Alert_Important_Only
                            Else, if context is: {first}
                                Use paragraph format: DD_Alert_Important_First
                            Else, if context is: {last}
                                Use paragraph format: DD_Alert_Last
                            Else, if context is: {notfirst}
                                Use paragraph format: DD_Alert_Mid
                    Else, if context is: Alert[AlertType = "Note"]

                        1.1.2.If context is: {only}
                                Use paragraph format: DD_Alert_Note_Only
                            Else, if context is: {first}
                                Use paragraph format: DD_Alert_Note_First
                            Else, if context is: {last}
                                Use paragraph format: DD_Alert_Last
                            Else, if context is: {notfirst}
                                Use paragraph format: DD_Alert_Mid
                    Else, if context is: Alert[AlertType = "Caution"]

                        1.1.3.If context is: {only}
                                Use paragraph format: DD_Alert_Caution_Only
                            Else, if context is: {first}
                                Use paragraph format: DD_Alert_Caution_First
                            Else, if context is: {last}
                                Use paragraph format: DD_Alert_Last
                            Else, if context is: {notfirst}
                                Use paragraph format: DD_Alert_Mid

                    Else, if context is: Alert[AlertType = "Warning"]

                        1.1.4.If context is: {only}
                                Use paragraph format: DD_Alert_Warning_Only
                            Else, if context is: {first}
                                Use paragraph format: DD_Alert_Warning_First
                            Else, if context is: {last}
                                Use paragraph format: DD_Alert_Last
                            Else, if context is: {notfirst}
                                Use paragraph format: DD_Alert_Mid
                    Else, if context is: (BulletedList | NumberedList)
                        1.1.5.Count ancestors named:  BulletedList, NumberedList
                            If level is: 1
                                Use paragraph format: DD_Para_L1
                            Else, if level is: 2
                                Use paragraph format: DD_Para_L2
                            Else, if level is: 3
                                Use paragraph format: DD_Para_L3
                            Else
                                Use paragraph format: DD_Para_L3
                    Else
                        Use paragraph format: DD_Para

        Else, if context is: * < (SubSection[TopicType = "PopUp: Include Text"] |
        SubSection[TopicType = "PopUp: Exclude Text"])

            1.2. If context is: Alert[AlertType = "Important"]

                    1.2.1.If context is: {only}
                            Use paragraph format: PopUp_Alert_Important_Only
                        Else, if context is: {first}
                            Use paragraph format: PopUp_Alert_Important_First
                        Else, if context is: {last}
                            Use paragraph format: PopUp_Alert_Last
                        Else, if context is: {notfirst}
                            Use paragraph format: PopUp_Alert_Mid
                Else, if context is: Alert[AlertType = "Note"]

                    1.2.2.If context is: {only}
                            Use paragraph format: PopUp_Alert_Note_Only
                        Else, if context is: {first}
                            Use paragraph format: PopUp_Alert_Note_First
                        Else, if context is: {last}
                            Use paragraph format: PopUp_Alert_Last
                        Else, if context is: {notfirst}
                            Use paragraph format: PopUp_Alert_Mid
                Else, if context is: Alert[AlertType = "Caution"]

                    1.2.3.If context is: {only}
                            Use paragraph format: PopUp_Alert_Caution_Only
                        Else, if context is: {first}
                            Use paragraph format: PopUp_Alert_Caution_First
                        Else, if context is: {last}
                            Use paragraph format: PopUp_Alert_Last
                        Else, if context is: {notfirst}
                            Use paragraph format: PopUp_Alert_Mid
                Else, if context is: Alert[AlertType = "Warning"]

                    1.2.4.If context is: {only}
                            Use paragraph format: PopUp_Alert_Warning_Only
                        Else, if context is: {first}
                            Use paragraph format: PopUp_Alert_Warning_First
                        Else, if context is: {last}
                            Use paragraph format: PopUp_Alert_Last
                        Else, if context is: {notfirst}
                            Use paragraph format: PopUp_Alert_Mid
                Else, if context is: (BulletedList | NumberedList)
                    1.2.5.Count ancestors named:  BulletedList, NumberedList
                        If level is: 1
                            Use paragraph format: PopUp_Para_L1
                        Else, if level is: 2
                            Use paragraph format: PopUp_Para_L2
                        Else, if level is: 3
                            Use paragraph format: PopUp_Para_L3
                        Else
                            Use paragraph format: PopUp_Para_L3
                Else, if context is: Alert

                    1.2.6.If context is: {only}
                            Use paragraph format: PopUp_Alert
                        Else, if context is: {first}
                            Use paragraph format: PopUp_Alert_First
                        Else, if context is: {last}
                            Use paragraph format: PopUp_Alert_Last
                        Else, if context is: {notfirst}
                            Use paragraph format: PopUp_Alert_Continued
                Else
                    Use paragraph format: PopUp_Para
```

Figure 60: Assigning paragraph tags to the Paragraph element

- At the outermost level, the context rule tests whether a Para element is in
    - A drop-down SubSection that starts open or closed
    - A pop-up SubSection that includes or excludes text
    - A table cell
    - A list
    - None of the above

▶ Tagging the Table Title (TblTitle) Element

Figure 61shows the all contexts rule that assigns a paragraph tag to a table title. This rule is simple because a table title only occurs in one context in a document and is represented by one paragraph style in the WebWorks template.

```
Element (Table Title): TblTitle
    General rule:   (<TEXT> | Index)*

    Attribute list
        1.  Name: ID                    Unique ID           Required
            Control flags:  Read-only,  Hidden

    Text format rules

        1.  In all contexts.
                Use paragraph format: TableTitle
```

Figure 61: Assigning a paragraph tag for the TblTitle element

```
Else, if context is: * < TblCell

    1.3. If context is: * < TblCell[HorizontalAlignment = "Left"]

            1.3.1.If context is: * < TblCell[VerticalAlignment = “Top”]
                    Use paragraph format: CellBody, LT
                Else, if context is: * < TblCell[VerticalAlignment = “Middle”]
                    Use paragraph format: CellBody, LM
                Else, if context is: * < TblCell[VerticalAlignment = “Bottom”]
                    Use paragraph format: CellBody, LB
                Else
                    Use paragraph format: CellBody, LT
        Else, if context is: * < TblCell[HorizontalAlignment = “Center”]

            1.3.2.If context is: * < TblCell[VerticalAlignment = “Top”]
                    Use paragraph format: CellBody, CT
                Else, if context is: * < TblCell[VerticalAlignment = “Middle”]
                    Use paragraph format: CellBody, CM
                Else, if context is: * < TblCell[VerticalAlignment = “Bottom”]
                    Use paragraph format: CellBody, CB
                Else
                    Use paragraph format: CellBody, CT
        Else, if context is: * < TblCell[HorizontalAlignment = “Right”]

            1.3.3.If context is: * < TblCell[VerticalAlignment = “Top”]
                    Use paragraph format: CellBody, RT
                Else, if context is: * < TblCell[VerticalAlignment = “Middle”]
                    Use paragraph format: CellBody, RM
                Else, if context is: * < TblCell[VerticalAlignment = “Bottom”]
                    Use paragraph format: CellBody, RB
                Else
                    Use paragraph format: CellBody, RT
        Else

            1.3.4.If context is: * < TblCell[VerticalAlignment = “Top”]
                    Use paragraph format: CellBody, LT
                Else, if context is: * < TblCell[VerticalAlignment = “Middle”]
                    Use paragraph format: CellBody, LM
                Else, if context is: * < TblCell[VerticalAlignment = “Bottom”]
                    Use paragraph format: CellBody, LB
                Else
                    Use paragraph format: CellBody, LT
```

```
        Else, if context is: Alert[AlertType = “Important”]

            1.4. If context is: {only}
                    Use paragraph format: Alert_Important_Only
                Else, if context is: {first}
                    Use paragraph format: Alert_Important_First
                Else, if context is: {last}
                    Use paragraph format: Alert_Last
                Else, if context is: {notfirst}
                    Use paragraph format: Alert_Mid
        Else, if context is: Alert[AlertType = “Note”]

            1.5. If context is: {only}
                    Use paragraph format: Alert_Note_Only
                Else, if context is: {first}
                    Use paragraph format: Alert_Note_First
                Else, if context is: {last}
                    Use paragraph format: Alert_Last
                Else, if context is: {notfirst}
                    Use paragraph format: Alert_Mid
        Else, if context is: Alert[AlertType = “Caution”]

            1.6. If context is: {only}
                    Use paragraph format: Alert_Caution_Only
                Else, if context is: {first}
                    Use paragraph format: Alert_Caution_First
                Else, if context is: {last}
                    Use paragraph format: Alert_Last
                Else, if context is: {notfirst}
                    Use paragraph format: Alert_Mid
        Else, if context is: Alert[AlertType = “Warning”]

            1.7. If context is: {only}
                    Use paragraph format: Alert_Warning_Only
                Else, if context is: {first}
                    Use paragraph format: Alert_Warning_First
                Else, if context is: {last}
                    Use paragraph format: Alert_Last
                Else, if context is: {notfirst}
                    Use paragraph format: Alert_Mid

    Else, if context is: (BulletedList | NumberedList)
        1.8. Count ancestors named:    BulletedList, NumberedList
            If level is: 1
                Use paragraph format: Para_L1
            Else, if level is: 2
                Use paragraph format: Para_L2
            Else, if level is: 3
                Use paragraph format: Para_L3
            Else
                Use paragraph format: Para_L3
    Else
        Use paragraph format: Para
```

Assigning paragraph tags to the Paragraph element, continued

▶ Tagging the Table Cell (TblCell) Element

Figure 62 shows the context rule used to assign paragraph tags to paragraphs that reside in table cells.

When we created the EDD and page template we decided we wanted greater control over the horizontal and vertical alignment of paragraphs in table cells. Thus we added attributes to the TblCell element and choices for aligning text horizontally (Left, Center, and Right) and vertically (Top, Middle, and Bottom). We then created paragraph styles in the WebWorks template that would implement the same level of control over alignment in the resulting help system. Consequently the EDD assigns 18 different paragraph tags to the paragraphs associated with table cells. These paragraph styles differentiate between heading cells and body cells, as well as implement all of the alignment options.

▶ Tagging the FigCaption Element

Figure 63 shows the all contexts rule that assigns a paragraph tag to a figure caption. This is another straightforward rule because figure captions only occur in one context within a document and the resulting help system.

```
Element (Container): FigCaption
    General rule:   (<TEXT> | Index)*

    Attribute list
        1.  Name: ID                          Unique ID             Required
            Control flags:  Read-only,  Hidden

    Text format rules

        1.  In all contexts.
                Use paragraph format: FigCaption
```

Figure 63: Assigning a paragraph tag to the FigCaption element

```
Element (Table Cell): TblCell
    General rule:   (<TEXT> | FigFrame | XRef | Index)* | (Para | NumberedList | BulletedList |
                    Figure)*

    Attribute list
        1.  Name: HorizontalAlignment       Choice                Optional
            Choices:    Left, Center, Right
        2.  Name: VerticalAlignment         Choice                Optional
            Choices:    Top, Middle, Bottom

    Text format rules

        1.  If context is: HeadingRow

                1.1. If context is: [HorizontalAlignment = “Left”]

                        1.1.1.If context is: [VerticalAlignment = “Top”]
                                Use paragraph format: CellHeading, LT

                            Else, if context is: [VerticalAlignment = “Middle”]
                                Use paragraph format: CellHeading, LM

                            Else, if context is: [VerticalAlignment = “Bottom”]
                                Use paragraph format: CellHeading, LB

                            Else
                                Use paragraph format: CellHeading, LT
                    Else, if context is: [HorizontalAlignment = “Center”]

                        1.1.2.If context is: [VerticalAlignment = “Top”]
                                Use paragraph format: CellHeading, CT

                            Else, if context is: [VerticalAlignment = “Middle”]
                                Use paragraph format: CellHeading, CM

                            Else, if context is: [VerticalAlignment = “Bottom”]
                                Use paragraph format: CellHeading, CB

                            Else
                                Use paragraph format: CellHeading, CT
                    Else, if context is: [HorizontalAlignment = “Right”]

                        1.1.3.If context is: [VerticalAlignment = “Top”]
                                Use paragraph format: CellHeading, RT

                            Else, if context is: [VerticalAlignment = “Middle”]
                                Use paragraph format: CellHeading, RM

                            Else, if context is: [VerticalAlignment = “Bottom”]
                                Use paragraph format: CellHeading, RB

                            Else
                                Use paragraph format: CellHeading, RT

                    Else

                        1.1.4.If context is: [VerticalAlignment = “Top”]
                                Use paragraph format: CellHeading, LT

                            Else, if context is: [VerticalAlignment = “Middle”]
                                Use paragraph format: CellHeading, LM

                            Else, if context is: [VerticalAlignment = “Bottom”]
                                Use paragraph format: CellHeading, LB

                            Else
                                Use paragraph format: CellHeading, CT

        Else

            1.2. If context is: [HorizontalAlignment = "Left"]

                    1.2.1.If context is: [VerticalAlignment = “Top”]
                            Use paragraph format: CellBody, LT

                        Else, if context is: [VerticalAlignment = “Middle”]
                            Use paragraph format: CellBody, LM

                        Else, if context is: [VerticalAlignment = “Bottom”]
                            Use paragraph format: CellBody, LB

                        Else
                            Use paragraph format: CellBody, LT
                Else, if context is: [HorizontalAlignment = “Center”]

                    1.2.2.If context is: [VerticalAlignment = “Top”]
                            Use paragraph format: CellBody, CT

                        Else, if context is: [VerticalAlignment = “Middle”]
                            Use paragraph format: CellBody, CM

                        Else, if context is: [VerticalAlignment = “Bottom”]
                            Use paragraph format: CellBody, CB

                        Else
                            Use paragraph format: CellBody, CT
                Else, if context is: [HorizontalAlignment = “Right”]

                    1.2.3.If context is: [VerticalAlignment = “Top”]
                            Use paragraph format: CellBody, RT

                        Else, if context is: [VerticalAlignment = “Middle”]
                            Use paragraph format: CellBody, RM

                        Else, if context is: [VerticalAlignment = “Bottom”]
                            Use paragraph format: CellBody, RB

                        Else
                            Use paragraph format: CellBody, RT

                Else

                    1.2.4.If context is: [VerticalAlignment = “Top”]
                            Use paragraph format: CellBody, LT

                        Else, if context is: [VerticalAlignment = “Middle”]
                            Use paragraph format: CellBody, LM

                        Else, if context is: [VerticalAlignment = “Bottom”]
                            Use paragraph format: CellBody, LB

                        Else
                            Use paragraph format: CellBody, LT
```

Figure 62: Assigning paragraph tags to the TblCell element

Stage 3:
Creating Rules that Format Documents for Print

At this stage, any FrameMaker document created with the sample EDD and page template will align perfectly with the sample WebWorks ePublisher Pro template. FrameMaker will automatically assign all of the paragraph styles necessary to generate the help system. An author need only create a WebWorks project based on the sample template, drag in some FrameMaker files, and generate the help system.

The final step in developing the EDD is to add the context formatting rules that format documents for print.

The sample page template and underlying EDD produces fully formatted FrameMaker documents. You can use the EDD and page template as is, or modify these files per your own project requirements.

The process for adding context rules that format the text is identical to the process used to assign the paragraph styles:

- Always begin the context rules that format the text AFTER the context rule that assigns the paragraph tags.
- Always work from the most specific case to the most general case.
- Follow a methodical approach for applying contextual formats to your content elements. We recommend the following:
    - The first rule should specify paragraph tags that align with the WebWorks template.
    - The second rule should specify paragraph indents and tabs for every context an element can be used in a document.
    - The third rule should apply spacing above and below the paragraph for every context an element can be used in a document.
    - The fourth rule should specify font family, size, and weight for every context an element can be used in a document.
    - The fifth rule should specify any advanced paragraph properties for every context an element can be used in a document.