



# Contents

---

<b>Developing Online Documents with FrameMaker 4</b> .....	<b>1</b>
Introduction to FrameMaker 4 and FrameViewer 41	
<b>How Hypertext Works</b> .....	<b>1</b>
Planning an Online Document2	
<b>What Types of Documents Should go Online?</b> .....	<b>2</b>
<b>Audience Considerations</b> .....	<b>2</b>
<b>System Planning.</b> .....	<b>3</b>
Mechanics of Creating Hypertext within FrameMaker 44	
<b>Where to Place Hypertext Markers</b> .....	<b>4</b>
<b>Indicating an Active Area in Text</b> .....	<b>4</b>
<b>Creating Hypertext in a Graphic</b> .....	<b>5</b>
<b>Creating Hypertext in the Same Place on Several Pages.</b> .....	<b>5</b>
<b>Inserting Hypertext Markers and Commands</b> .....	<b>5</b>
<b>Types of Hypertext Commands.</b> .....	<b>6</b>
Mechanics of Automatically Generating Hypertext15	
<b>Display Commands for Generated Hypertext Documents</b> .....	<b>15</b>
Activating Hypertext Within FrameMaker16	
<b>Locking a Document</b> .....	<b>16</b>
<b>Unlocking a Document.</b> .....	<b>16</b>
<b>Creating Nonprinting Buttons.</b> .....	<b>16</b>
Advanced Features of Hypertext16	
<b>Setting View-Only Document Options</b> .....	<b>16</b>
<b>Changing the Document Window</b> .....	<b>17</b>
<b>Using Active Cross-References</b> .....	<b>18</b>
<b>Disabling Commands.</b> .....	<b>18</b>
Moving Hypertext Documents from FrameMaker 3 to 419	
<b>Hypertext Changes.</b> .....	<b>19</b>
<b>Using FrameViewer on Multiple Platforms</b> .....	<b>21</b>
Cross-Platform Issues21	
<b>Pathname Conventions</b> .....	<b>21</b>
<b>Graphic Formats.</b> .....	<b>21</b>
Context-Sensitive Help22	
Unix Specific FrameViewer Information22	
<b>Entering Command-Line Options for FrameViewer 4 for Unix</b> .....	<b>22</b>
<b>Specifying a File to Open on Start-Up</b> .....	<b>22</b>
<b>Minimizing the Unix FrameViewer Installation</b> .....	<b>24</b>

---

# Contents

---

<b>Licensing</b> .....	<b>25</b>
<b>Context-Sensitive Help for Unix</b> .....	<b>26</b>
Windows Specific FrameViewer Information	27
<b>Context-Sensitive Help for Windows</b> .....	<b>27</b>
Macintosh Specific FrameViewer Information	28
<b>Distribution/CD-ROM Production</b> .....	<b>29</b>
CD-ROM Technology	29
<b>Why Manufacture a CD-ROM?</b> .....	<b>29</b>
<b>Types of Data Stored on CD-ROM</b> .....	<b>29</b>
<b>Updating CD-ROM Discs</b> .....	<b>29</b>
<b>How Much Does It Cost?</b> .....	<b>29</b>
<b>CD-ROM Development and Production Processes</b> .....	<b>30</b>
<b>CD-ROM Manufacturing Facilities</b> .....	<b>31</b>
Unix Distribution Issues	32
MS Windows Distribution Issues	33
<b>Uninstalled Application</b> .....	<b>33</b>
<b>Installed Application Uncustomized</b> .....	<b>33</b>
<b>Installed Application Customized</b> .....	<b>34</b>
<b>Troubleshooting</b> .....	<b>37</b>

---

---

# 1

## ***Developing Online Documents with FrameMaker 4***

Revision 2 — July 27, 2010

---

### **Section 1-1**

### **Introduction to FrameMaker 4 and FrameViewer 4**

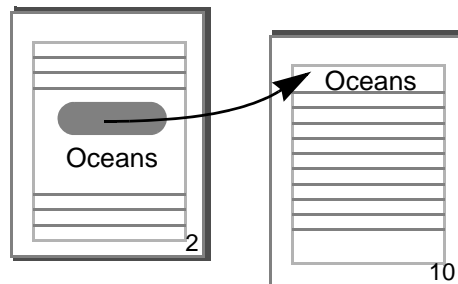
---

This document is intended to provide FrameMaker 4 user's issues and information related to designing online documents. Much of the information contained in this document can be found in following FrameMaker 4 manuals: *Using FrameMaker*, *MIF Reference*, *Changing Setup Files*, and *Release Notes*.

FrameMaker allows you to create sophisticated, view-only documents — perfect for communication in an online environment using FrameViewer. FrameMaker writers can both manually and automatically insert hypertext within documents so viewers can instantly review related pieces of information. For example, a viewer can click on an illustration in an online parts manual and view the inventory status for that particular product. Or, they can click on a photo in a newsletter and immediately browse through document archives related to the photo image.

#### **How Hypertext Works**

In online documents, you create hypertext links (active areas) between related pieces of information as the document writer. Then, when a viewer clicks a hypertext link (a word, phrase, or graphic containing a hypertext command) in FrameViewer (or a view-only FrameMaker document) the command is carried out. For example, the command may display another page in the same document or in a different document, a pop-up menu of topics, or an alert box containing a few lines of information.



**FIGURE 1. Example of Using Hypertext**

---

## Section 1-2

## Planning an Online Document

---

### What Types of Documents Should go Online?<sup>1</sup>

There are no absolute guidelines to determine what types of documents should go online, however, consider the following: cost and value. Will it be more expensive to produce online documents or printed documents? Do the documents add inherent value to your business? The following documents are good candidates for use online:

- Urgent Information Documents
  - Material Safety Data Sheets
  - Technical Notices
  - Recall Notices
  - Engineering Change Notices
  - Software Change Notices
- Evolving Documents
  - Change Pages
  - Sales Catalogs
  - Price Lists
  - Procedure Manuals
  - Errata Sheets
  - Supplements
  - Replacement Chapters or Sections
- Inter-related Information
  - Cross-Referenced Information (i.e., See Figure 2 on page 9)
  - Repeated Data (i.e., technical data that references the same figure throughout the entire document)
  - Related Documents (i.e, See the Windows Reference Manual for additional information)

Not all documents are good candidates for use online. For example, documents that contain lengthy and detailed text (i.e., abstract philosophy) may be more easily read in hard-copy. In addition, legal documents and those documents needed away from the computer, as well as documents for non-computer users, may not be good candidates for use online.

### Audience Considerations

There are four primary areas to consider when planning an online document:

1. Types of users
2. Tasks that the users will perform
3. How the users will interact with the online document
4. Types of documents the users will navigate

---

1. See "Designing & Writing Online Documentation" by William K. Horton for additional information.

## System Planning

After you have considered all the issues pertinent to your online audience, strategize about the system planning. For example, consider the size of each document. The viewer can display each page within a long document quickly; however, a long document takes more time to open. With a series of short documents (typically 10 pages or less), there is little delay when a viewer clicks an active area to display a page from another document.

Also, consider whether you will open several document windows. If so, is it important to be able to retrace your jumps? Remember, each new open window starts its own stack for retracing jumps, and the two do not merge. See “Types of Hypertext Commands” on page 6.

**Determine the Method for Delivering Online Documents.** When planning an online document, consider the following areas:

1. What is the smallest screen size available to view these documents?  
Choose a document size that displays best on all monitors. For example, an 8.5” by 5.5” page size works well for most monitors, even 13” monitors.
2. What type of color capability will be available on the end-user systems?  
Choose colors that display well on all systems (i.e., color tints do not display well on lower dpi monitors). Use the lowest common denominator computer (i.e., slowest) when developing an online system if possible.  
Consider using a color and underline FrameMaker Character Format if the end-user system involves both color and monochrome monitors.
3. Are the online documents for the help system stand-alone or context-sensitive?  
Context-sensitive help systems require more extensive planning. Every location within your application must be identified where FrameViewer will be accessed. Once these areas have been identified, the Frame Developer’s Kit (FDK) can be used to create context-sensitive access. (See “Context-Sensitive Help” on page 22.)
4. What fonts will be available on the end-user system?
5. Will both paper and view-only versions be issued simultaneously?  
If both paper and view-only versions will be issued simultaneously, consider using FrameMaker’s Conditional Text for the various versions.

**Resolve Cross-Platform Issues.** If your online system will be accessed from various hardware platforms, consider the following:

- Use the 8.3 monospace filename convention for all files and directories.
- Use graphic file formats that display across all hardware platforms that the end-user will be using (i.e., TIFF file format).
- Use fonts that are available on all end-user platforms (i.e., Times and Helvetica). Remember that unique symbols (i.e., Zapf Dingbats) used within the document may not be the same on all hardware platforms.
- Use Unix pathname conventions for specifying the pathname of a file if you are planning online documents for Unix and/or Macintosh and Windows.

**Resolve Graphics Issues.**

- If possible, limit the number of graphics.
- Line art is more readable on screen but may be harder to maintain.

- If you use screen shots, convert to monochrome if possible.
- Importing graphics by reference is safer (graphics don't get lost) but can create additional file management issues.

**Storyboard the Process.**

1. Mockup a paper document that indicates the hypertext areas.  
This will help eliminate unnecessary hypertext as well as point out deficient areas (i.e., where a viewer may want to navigate but cannot).
2. Design and test an online document prototype.
3. Define and test the online document production process.
4. Create a schedule based on production process taking into account hypertext link management.
5. Determine how and when the documents will be updated.
6. Does the online version need to be paginated in the same way as the paper version?
7. Determine how many hypertext markers can be automatically generated.  
Hypertext markers can be automatically generated when you generate a Table of Contents, Index, List of Figures, List of Tables, etc. In addition, all cross-references create hypertext markers in FrameViewer.

---

**Section 1-3**

**Mechanics of Creating Hypertext within FrameMaker 4**

---

**Where to Place Hypertext Markers<sup>2</sup>**

You can create hypertext in the following areas within FrameMaker 4 documents:

- Text — so viewers can click a word, phrase, or sentence to see related information or hear related messages.
- Graphics — so viewers can click a portion of a drawing for more information.
- The same place on several pages — so viewers can easily retrace their path through the document(s) or page through an online document.

**Indicating an Active Area in Text**

To use a word, phrase, or paragraph to indicate an active area, select the text, change its Character Format, and insert a hypertext marker. Choose a Character Format such as red and underline to visually indicate a hypertext area to the viewer. Then, when a viewer clicks in a text column in a locked document, FrameViewer highlights the area and performs the hypertext command. FrameViewer ignores Character Formatting such as change bar, spread, case, and pair kerning. If you need to designate a word or phrase as hypertext without changing the character properties, use a Character Format that sets all properties to “As Is”.

If you insert hypertext markers at the beginning or end of words, text editing will be easier in the future.

---

2. For addition information, see Chapter 27 “Hypertext and View-Only Documents” in the *Using FrameMaker* Manual for FrameMaker 4.

To use an entire paragraph to indicate an active area, make sure there are no Character Format changes in the paragraph and insert a hypertext marker. If there are no Character Format changes in the paragraph, then when a viewer clicks, FrameViewer searches the paragraph until it finds a hypertext marker, highlights the entire paragraph, and then performs the hypertext command.

### Creating Hypertext in a Graphic

A hypertext marker must always be in a text column, therefore, you must put a text column on top of a graphic to create an active area. To create hypertext in a graphic, perform the following functions:

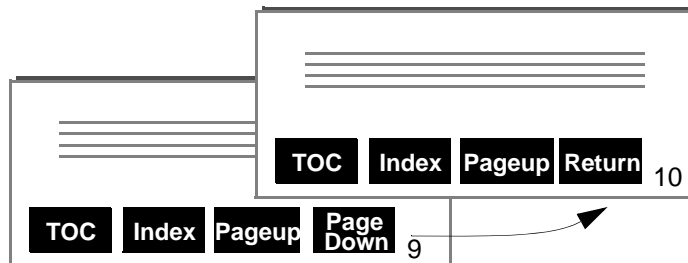
1. Draw a text column over the graphic you want to be active.
2. Set the Pen and Fill patterns to None.
3. Make sure that the text column is on top of the graphic.
4. Insert a hypertext marker.

With a hypertext marker present, the entire text column is an active area. When a viewer clicks anywhere in the text column, FrameViewer highlights the entire column.

### Creating Hypertext in the Same Place on Several Pages

When you want an active area to appear on several pages of a document, create the hypertext marker on a master page. When a viewer clicks that area on a Body Page, FrameViewer looks for a hypertext marker on the Master Page.

You can override the Master Page hypertext areas on any Body Page of the document. For example, you may want the last page of every document not to contain a pagedown button since it is the last page. To override a Master Page's active area, put an active area or an object over it on a Body Page.



### Inserting Hypertext Markers and Commands

After you indicate hypertext areas in a document, you are ready to insert hypertext markers and commands. To insert:

1. Choose Marker from the Special menu. The Marker window will appear.
2. Put the insertion point where you want to insert a marker. Editing your document will be easier if you put markers at the beginning or end of a word.
3. In the Marker Type scroll list, select Hypertext.

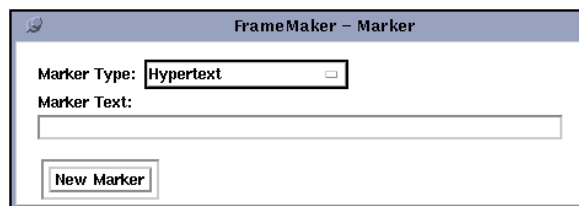


FIGURE 2. Selecting a Hypertext Marker in FrameMaker 4 for X/Motif

4. In the Marker Text box, type the hypertext command.  
When inserting hypertext markers in text, make sure that the hypertext marker is located with the Character Format change, otherwise the hypertext area will be before the active area. If you experience difficulty typing in characters into the Marker dialog box, use the cross-platform method of the “/” forward slash. (See “Pathname Conventions” on page 21.)

## Types of Hypertext Commands

Hypertext markers embed links or commands in a document. When the document is opened in FrameViewer (or as view-only in FrameMaker), these markers execute the specified commands. The marker text specifies the command. To locate these commands in FrameMaker 4 simply select the Help Main Menu ..., press Syntax Descriptions, then select Hypertext commands.


Important: FrameMaker/FrameViewer remembers the origins of a viewer’s last 69 jumps in the document window by storing them on a stack, with the most recent jump on top. The stack lets a viewer retrace his or her path. In the following descriptions, each command’s effect on the hypertext stack is explained. Listed below are the various types of hypertext commands that you can create with FrameMaker 4. Those commands listed with an asterisk affect the stack.

alert	gotolink*	gotolink firstpage*	gotolink lastpage*
gotopage*	matrix <sup>1</sup>	message	message system
newlink	nextpage*	openlink <sup>2</sup>	opennew
openpage <sup>2</sup>	popup <sup>3</sup>	previouslink <sup>4</sup>	previouspage*
quit	quitall		

1. Commands in the matrix have the same effect on the stack as they would elsewhere.
2. Creates a stack for the newly opened window, but leaves the stack in the current window unchanged.
3. Commands in the matrix have the same effect on the stack as they would elsewhere.
4. Displays the last location stored on the stack.




### Types of Hypertext Markers Available in FrameMaker 4

Hypertext Name	Purpose	Syntax	Example/Comments
alert	Displays an alert box.	<p>alert <i>message</i></p> <p><i>message</i> is the text that appears in the alert box.</p>	<p>The message can be up to 249 characters long.</p> 
<p>There are three types of gotolinks. The first type, <code>gotolink</code>, requires another marker to be placed in the document. This additional marker is a hypertext marker called <code>newlink</code>. See “<code>newlink</code>” on page 10. Gotolinks are different from <code>openlinks</code> because they do not open a separate document window.</p>			
gotolink	Displays the page that contains a matching <code>newlink</code> .	<p><code>gotolink filename:linkname</code></p> <p><i>filename</i>: identifies the file that contains the matching <code>newlink</code>. If the <code>newlink</code> is in the current file, <i>filename</i>: is not needed.</p> <p><i>linkname</i> is the name of the corresponding <u><code>newlink</code></u>.</p>	<p>This hypertext link jumps to the page of the specified document containing a <code>newlink</code> command with a matching linkname. For example, the command <code>gotolink Repair.doc:Brakes</code> causes FrameViewer to jump to the page (in the document <code>Repair.doc</code>) that contains the <code>newlink Brakes</code>.</p> <p>If FrameViewer can't find the specified document or linkname, it leaves the current page displayed.</p> <p>If you need to resize the document window, use the <code>openlink</code> command. The <code>gotolink</code> command does not resize the document window.</p>
<p>(Sheet 1 of 8)</p>			

## Types of Hypertext Markers Available in FrameMaker 4 (Continued)

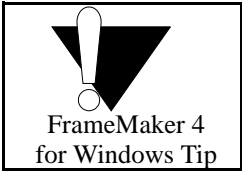
Hypertext Name	Purpose	Syntax	Example/Comments
gotolink firstpage	Displays the first page of the document.	<i>gotolink filename:firstpage</i> <i>filename</i> identifies a file other than the current one. To go to the first page of the current file, <i>filename:</i> is not needed.	The linkname in this command must match exactly the linkname in the newlink command. For example, a linkname with a trailing space differs from a linkname without a trailing space. Also, the linkname is case-sensitive; <i>Changing Units</i> differs from <i>changing units</i> .
gotolink lastpage	Displays the last page of the document.	<i>gotolink filename:lastpage</i> <i>filename</i> identifies a file other than the current one. To go to the last page of the current file, <i>filename:</i> is not needed.	The filename in a <i>gotolink</i> command can contain a pathname. When the pathname is relative, FrameViewer searches for the relative pathname in the following directories (in this order): <ul style="list-style-type: none"> <li>• The directory containing the document you're jumping from.</li> <li>• <i>install_dir/fmunit/usenglish/helpdir</i>.</li> </ul> where <i>install_dir</i> is the directory in which FrameViewer is installed. (See "ISSUE 5" on page 39.)
gotopage	Displays a specific page in the document.	<i>gotopage filename:pagenumber</i> <i>filename:</i> identifies a file other than the current one. To go to a specific page of the current file, <i>filename:</i> is not needed.  <i>pagenumber</i> is the number of the page to be displayed.	FrameViewer will beep as the first page of a document is displayed if the <i>pagenumber</i> for the command is invalid. The document that you are jumping to may not have page numbering starting at 1 (because it is part of a book where page numbering is continuous, or you have set the starting page to something other than 1). Therefore, using <i>gotopage filename:1</i> as your hypertext command is not correct. You should either edit your hypertext command to reflect the actual page number of the first page in the document or you should use the <i>gotolink filename:firstpage</i> command if you just want to go to the first page of that document (no matter what the actual page number is).

## Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments
matrix	Divides the text column containing the marker into rows and columns of hypertext buttons. The commands for these buttons are specified in a text column on a reference page.	<p><code>matrix rows columns flowname</code></p> <p><i>rows</i> specifies how many rows of buttons the text column will contain.</p> <p><i>columns</i> specifies the number of columns of buttons the text column will contain.</p> <p><i>flowname</i> is the name of the flow tag for the corresponding text column on the reference page.</p>	<p><code>matrix 2 3 MyCommands</code> divides the text column into two rows and three columns of buttons. The hypertext commands are on a reference page in a text column tagged <i>MyCommands</i>. Each command is in a separate paragraph: the first three paragraphs are for row 1, and the second three paragraphs are for row 2.</p> <p>The example below depicts where to use the matrix hypertext command in a graphic. This type of graphic lends itself to the matrix command because each figure is the same size and the figures are divided into rows and columns.</p> 
message	Sends a message to another application that's already running.	<p><code>message apiclient yourmessage</code></p> <p><i>apiclient</i> is the registered name of the API application.</p> <p><i>yourmessage</i> is the text passed to the API application.</p>	

(Sheet 3 of 8)

## Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments										
message system	Executes a platform-specific command.	<p>message system <i>command</i></p> <p><i>Command</i> is the command name, as you would type it on any platform.</p>	<p>You can start other applications from within FrameViewer (or a view-only FrameMaker document) using the message system hypertext command. However, the message system command is not cross-platform compatible (i.e., a Windows hypertext message commands will not work on the Unix platform).</p> <div data-bbox="1444 423 1684 594" style="text-align: center;">  <p>FrameMaker 4 for Windows Tip</p> </div> <p>Under FrameMaker4 for Windows, the following commands can be used with the message system command. The command is as follows: message system application pathname, winstate where winstate is one of the following<sup>1</sup>:</p> <table border="0" style="width: 100%;"> <tr> <td>SW_HIDE</td> <td>SW_MINIMIZE</td> </tr> <tr> <td>SW_RESTORE</td> <td>SW_SHOW</td> </tr> <tr> <td>SW_SHOWMAXIMIZED</td> <td>SW_SHOWMINIMIZE</td> </tr> <tr> <td>SW_SHWOMINNOACTIVE</td> <td>SW_SHOWNA</td> </tr> <tr> <td>SW_SHOWNOACTIVE</td> <td>SW_SHOWNORMAL</td> </tr> </table>	SW_HIDE	SW_MINIMIZE	SW_RESTORE	SW_SHOW	SW_SHOWMAXIMIZED	SW_SHOWMINIMIZE	SW_SHWOMINNOACTIVE	SW_SHOWNA	SW_SHOWNOACTIVE	SW_SHOWNORMAL
SW_HIDE	SW_MINIMIZE												
SW_RESTORE	SW_SHOW												
SW_SHOWMAXIMIZED	SW_SHOWMINIMIZE												
SW_SHWOMINNOACTIVE	SW_SHOWNA												
SW_SHOWNOACTIVE	SW_SHOWNORMAL												
newlink	Identifies a destination for gotolink and openlink commands.	<p>newlink <i>linkname</i></p> <p><i>linkname</i> is a string that uniquely identifies the newlink. It is the target for <u>openlink</u> and <u>gotolink</u> commands that use the same <i>linkname</i>.</p>	If the link is in a heading, you may want to use the heading name as the linkname to simplify the maintenance of your documents.										
nextpage	Displays the next page of the current document.	nextpage											

(Sheet 4 of 8)

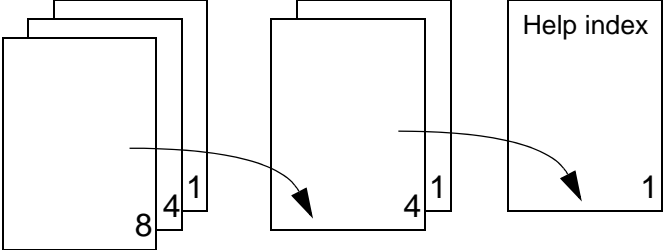
### Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments
openlink	Opens a new document window and displays the page that contains a matching newlink.	<p><code>openlink filename:linkname</code></p> <p><i>filename</i> identifies the file that contains the matching newlink. If the newlink is in the current file, <i>filename:</i> is not needed.</p> <p><i>linkname</i> is the name of the corresponding <u>newlink</u>.</p>	<p>Use the <code>openlink</code> command to display related information without losing the source. For example, clicking on the words Main Index opens the main index menu to FrameMaker's online help system. <code>Openlink</code> can open locked or unlocked documents.</p> <p>When FrameViewer jumps to an <code>openlink</code> command, it creates a stack for the newly opened window, but leaves the stack in the current window unchanged.</p>
opennew	Opens the specified file as an untitled document in a new document window.	<p><code>opennew filename</code></p> <p><i>filename</i> identifies the file to be opened.</p>	
openpage	Opens a new document window and displays a specific page in the document.	<p><code>openpage filename:pagenumber</code></p> <p><i>filename</i> identifies a file other than the current one. To go to a specific page of the current file, <i>filename:</i> is not needed.</p> <p><i>pagenumber</i> is the page number of the page to display.</p>	
(Sheet 5 of 8)			

## Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments
popup	Displays a pop-up menu whose items execute hypertext commands. The menu items are specified in a text column on a reference page. FrameViewer for the Unix platforms will only show you 24 items per pull down. There is not a limitation of FrameViewer for the Macintosh	<p style="text-align: center;"><code>popup flowname</code></p> <p><i>flowname</i> is the name of the flow tag for the corresponding text column on the reference page.</p>	<p><code>popup MyCommands</code> displays a popup menu. The items in the pop-up menu are specified in a text column tagged <i>MyCommands</i>. Each item is specified by a separate paragraph. The hypertext command for each item is specified by inserting a hypertext marker into its corresponding paragraph.</p> <p>For example, to create a pop-up menu used in the document to access the hypertext commands (below):</p> <ol style="list-style-type: none"> <li>1. Create an active area to be associated with the pop-up menu.</li> <li>2. Insert a hypertext marker containing a <code>popup</code> command.</li> <li>3. On a reference page, draw a text column and use the Flow command to assign a tag that matches the name in the popup command.</li> <li>4. In the Flow, type the menu name and press Return. In this example, the menu name is <code>HypertextCommands</code>.</li> <li>5. In each subsequent paragraph, type a menu item.</li> </ol> <p>If necessary, widen the text column so each item fits on a single line. The flow tag cannot contain embedded spaces, for example, <code>popup HypertextCommands</code>.</p> <p>Insert a hypertext marker in each line except the line with the menu name. The text column containing the menu items and markers would look something like this:</p> <pre style="border: 1px dotted black; padding: 5px;"> HypertextCommands alert gotolink matrix message system newlink nextpage openlink popup previouslink quit quitall </pre>

### Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments
popup (con't)			<p>When a viewer points on an active area containing the popup command and holds down the left mouse button, FrameViewer displays the following pop-up menu:</p> <div data-bbox="1383 402 1734 875" style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p><b>alert</b></p> <p><b>gotolink</b></p> <p><b>matrix</b></p> <p><b>message system</b></p> <p><b>newlink</b></p> <p><b>nextpage</b></p> <p><b>openlink</b></p> <p><b>popup</b></p> <p><b>previouslink</b></p> <p><b>quit</b></p> <p><b>quitall</b></p> </div> <p>When a viewer chooses an item from the menu, FrameViewer executes the corresponding hypertext command and changes the stack as needed.</p>
previouslink	Displays the last page the viewer viewed.	previouslink	

(Sheet 7 of 8)

### Types of Hypertext Markers Available in FrameMaker 4 (Continued)

Hypertext Name	Purpose	Syntax	Example/Comments
previouspage	Displays the previous page of the current document.	previouspage	
quit	Removes the current window from the screen.	quit	The <code>quit</code> command is the same as the Close command in the Window menu. If you edited the document and locked it without saving changes, FrameMaker prompts you to save it before closing.  If several locked documents are open, quit removes only the current locked document. To remove all locked documents, use the <code>quitall</code> command.
quitall	Removes all view-only windows from the screen, including documents that are closed to an icon.	quitall	
<b>(Sheet 8 of 8)</b>			

1. See the Power of... FrameMaker 4.0 for Windows, Chapter 23 by David B. Doty for additional information.



---

## Section 1-4      Mechanics of Automatically Generating Hypertext

---

### Display Commands for Generated Hypertext Documents

Hypertext markers can be automatically generated when you generate a Table of Contents, Index, List of Figures, List of Tables, etc. In addition, all cross-references create hypertext markers in FrameViewer. (See “Advanced Features of Hypertext” on page 16.) This features can significantly reduce the development process for creating online documents.

When the viewer clicks an active area in a generated hypertext document, FrameViewer displays the correct page in the source document. For best results, avoid using a Paragraph Tag with the “Frame Above” or “Frame Below” on when automatically generating hypertext. The following table lists some of the hypertext commands you can use for controlling how the source document is displayed. (The `gotoObjectId` command is a special hypertext command used only in generated documents.) You should type the commands exactly as shown. Use the `<$relfilename>` building block to have FrameViewer create a relative pathname from the generated document to the other document. Otherwise, use `<$fullfilename>` for a full pathname. (The examples in the following table show only the `<$fullfilename>` building block.)

To display	Type <sup>1</sup>
The highlighted item centered in the current window	<code>gotoObjectId</code> <code>&lt;\$fullfilename&gt;:&lt;\$ObjectType&gt;</code> <code>&lt;\$ObjectId&gt;</code>
The highlighted item centered in a new window	<code>openObjectId</code> <code>&lt;\$fullfilename&gt;:&lt;\$ObjectType&gt;</code> <code>&lt;\$ObjectId&gt;</code>
The page in the current window with nothing highlighted	<code>gotopage</code> <code>&lt;\$fullfilename&gt;:&lt;\$pagenumonly&gt;<sup>2</sup></code>
The page in a new window with nothing highlighted	<code>openpage</code> <code>&lt;\$fullfilename&gt;:&lt;\$pagenumonly&gt;<sup>2</sup></code>
An alert that displays the filename for the source document	<code>alert &lt;\$fullfilename&gt;</code>

1. Do not apply Character Formats to the text when typing these commands.
2. Avoid using the `<$pagenumonly>` building block if your page numbers are alphabetic or Roman characters. If your page numbers are alphabetic or Roman use the `<$ObjectId>` building block instead.

---

## Section 1-5

## Activating Hypertext Within FrameMaker

---

### Locking a Document

You can lock a document to test hypertext commands within FrameMaker.

Because your viewers see the locked document exactly as it appeared when you locked it, you may want to turn off the display of the grid, ruler, text symbols, and borders before locking it. Also, make sure that nothing is selected when you lock a document; otherwise, the selected object's handles will appear when a viewer opens the locked document.

To save a document in locked format, use the Save As command and turn on View-Only Format in the Save As dialog box.

To lock a document without saving it, press Esc F l k (for File lock). The command is case-sensitive, so be sure to type the letters as shown here. On the Macintosh, you can also press Command-Option-3. This is a toggle command, therefore repeat the procedure to change the document back to editable format.

In addition, hypertext commands can be activated without changing the document to View-Only format by using one of the following commands:

Windows	Alt-Control-click a hypertext marker
Macintosh	Control-Option-click a hypertext marker
X/Motif	Control-click right mouse button on a hypertext marker

### Unlocking a Document

To edit a locked document, you must unlock it first. When the document is unlocked, hypertext commands are inactive.

To unlock a document, press Esc F l k. The command is case-sensitive, so be sure to type the letters as shown here.

### Creating Nonprinting Buttons

You can create navigation buttons and other such graphics that won't print if your viewer prints the document. To make a graphic nonprinting, assign a color to the graphic and then put that color in the Don't Print list in the Set Up Separations dialog box. In the same dialog box, turn on Skip Blank Separation Pages. Make sure Print Separations is turned on in the Print dialog box and that you click Save Setting Only. For more information on making a color nonprinting and for printing separations, see Chapter 17 in the *Using FrameMaker* Manual for FrameMaker 4.

---

## Section 1-6

## Advanced Features of Hypertext

---

### Setting View-Only Document Options<sup>3</sup>

You can use MIF statements to control the display, appearance, and behavior of View-Only documents and cross-references in locked documents.

---

3. See the *MIF Reference* Manual under online Manuals in the FrameMaker 4 Help for additional information.

The MIF statements for View-Only documents are intended for hypertext authors who want more control over hypertext documents. They do not have corresponding commands in the user interface.

The View-Only MIF statements described in this section must appear in a <Document> statement. These statements have no effect in an unlocked document. Make sure that the <Document> statement also includes the following substatement:

```
<DViewOnly Yes>
```

To permanently lock a document perform the following commands:

1. Save the document out as MIF.
2. Open the MIF text file using whatever editor is available to you (in Unix press the shift key while opening the MIF file in FrameMaker and this will open the document as text only).
3. Search for DViewOnly
4. In the line directly below the DViewOnly statement type the following:

```
<DViewOnlyWinNoOp 0xF00>
```

5. Save, and quit the MIF file.
6. In FrameMaker, open this document.
7. Using the keyboard shortcut, Esc-F-l-k, lock the document. You will be presented with an alert telling you that this will now become a permanently locked document, and offer to make a backup that is ununlockable for future editing.

**WARNING: Always create a backup version of your document for future editing! Frame Technology takes no responsibility for the safety nor the security of documents protected in the manner described above or like functions, nor does it take responsibility for information rendered inaccessible by these means or other like functions.**

## Changing the Document Window

You can use MIF statements to change the appearance and behavior of the document window in the following ways:

- To suppress the document window menu bar, use the following statement:

```
<DViewOnlyWinMenubar No>
```

This statement has no effect in the Macintosh and Microsoft Windows version of a Frame product because those versions have an application menu bar rather than a document window menu bar.

- To suppress the display of scroll bars and border buttons in the document window, use the following statement:

```
<DViewOnlyWinBorders No>
```

To suppress selection in the document window, include the following statement:

```
<DViewOnlySelect No>
```

- You can normally select text and objects in a locked document by Control-dragging. Specifying `<DViewOnlySelect No>` prevents all selection in a locked document.

To suppress the appearance of a document region pop-up menu, use the following statement:

```
<DViewOnlyWinPopup No>
```

A *document region pop-up menu* is a menu activated by the right mouse button. For example, in Unix versions of a Frame product, the Maker menu and Viewer menus can be accessed by pressing the right mouse button. If this statement has a value of `No`, the background menu does not appear when the right mouse button is pressed. This statement has no effect in Macintosh and Windows versions of a Frame product.

- To make a window behave as a palette window, use the following statement:

```
<DViewOnlyWinPalette Yes>
```

A *palette window* is a command window that has special behavior, such as the Equations palette. The behavior of palette windows is platform-dependent. In the Macintosh version of a Frame product, for example, a palette window always remains in front of the active window. In a Unix version, a palette window can only be dismissed and cannot be closed to an icon.

## Using Active Cross-References

A locked document automatically has active cross-references. An *active cross-reference* behaves like a hypertext `gotoLink` command; when the user clicks on a cross-reference, FrameViewer displays the link's destination page. By default, the destination page is shown in the same document window as the link's source.

You can use MIF statements to turn off active cross-references and to change the type of hypertext link that the cross-reference emulates.

To make cross-references emulate the `openLink` command, which displays the destination page in a new document window, use the following statement:

```
<DViewOnlyXRef OpenBehavior>
```

Use this setting to allow users to see both the source page and the destination page.

To turn off active cross-references, use the following statement:

```
<DViewOnlyXRef NotActive>
```

Use this setting to emulate the behavior in earlier FrameMaker releases.

You can use the `<DViewOnlySelect>` statement to control whether active cross-references highlight the destination text. You might use this statement to improve the appearance of hypertext documents.

When cross-references are active and `<DViewOnlySelect Yes>` is specified, clicking a cross-reference in the document highlights the destination text. When cross-references are active and `<DViewOnlySelect UserOnly>` is specified, clicking a cross-reference does not highlight the destination text.

## Disabling Commands

In the Unix and Windows versions of FrameMaker, you can disable commands in a View-Only document. For example, a hypertext author might disable cut, copy, and print commands for sensitive documents.

To disable a command, you must supply the hex code, called an *fcode*, that internally represents that command in a Frame product. For example, you can disable printing, cutting, copying, and unlocking the document by supplying the following statements:

```
<DViewOnlyNoOp \0x313> # Disable printing  
<DViewOnlyNoOp \0x321> # Disable cutting or clearing  
<DViewOnlyNoOp \0x322> # Disable copying  
<DViewOnlyNoOp \0xF00> # Disable unlocking the document
```

The *fcodes* for commands are in the file *Commands*. You can find this file as follows:

For this Version of a Frame Product	Look Here
Unix	<code>\$FMHOME/fmunit/language</code> , where <i>language</i> is the language in use, such as <i>usenglish</i>
Windows	In the <i>fmunit</i> directory where the Frame product is installed

One undocumented hypertext command involving the use of *fcodes* is as follows:

*fcodes hexnumber*

The *hexnumber* is a function code listed in the above mentioned *Commands* file. For example, to completely exit FrameViewer (not just the document window), type the following in the hypertext marker window:

*fcodes fff* (This will exit the entire application.)

For more information about disabling commands, see the online manual *Changing Setup Files*<sup>4</sup>.

---

## Section 1-7

## Moving Hypertext Documents from FrameMaker 3 to 4

---

### Hypertext Changes<sup>5</sup>

There have been some modifications in the FrameMaker hypertext from version 3 to version 4. The changes to hypertext have the following effects:

- Graphics over buttons

In version 3 documents, if you use a text column on a Master Page as a button and if you have a graphic object that is on top of the text column, the button will work. In version 4 the button will no longer work. To fix the button, make sure the text column is on top of the graphic. Simply, select the graphic object and move it behind the text column. Change the text column's fill and pen patterns to None if necessary.

---

4. Note: The *fcodes* for disabling printing, cutting, copying, and unlocking in a document are incorrectly specified in the *MIF Reference Manual*.

5. See the *Release Notes* document under online Manuals in FrameMaker 4 Help for additional information.

- Newlink command highlighting

As a by-product of allowing selection in view-only documents, when you jump to a newlink, the marker is highlighted. This is useful if you use automatically generated links to debug an index. However, you might want to turn off this feature for documents you created in earlier versions.

To turn off the highlighting of newlink markers, import a MIF file that specifies how to handle selection in view-only documents. The MIF file should read as follows:

```
<MIFFile 4.0>
<Document
<DViewOnlySelect UserOnly>
# Turns off link selection but lets users select
>
```

Instead of importing a MIF statement, you can write your documents out to MIF and then edit the appropriate statements. (See “Setting View-Only Document Options” on page 16.)

#### Active area highlighting

In FrameMaker version 3, active areas in text were determined by placing a hypertext marker within the text and changing the Character Format of that text. If there were no Character Format changes, the entire paragraph would be active. Even if the paragraph included more than one hypertext marker, the entire paragraph would be active, and the first marker in the paragraph would take effect.

In version 4, you can still use Character Formats to specify active areas. However, when a paragraph has more than one hypertext marker, the paragraph is split by those markers into different active areas. The markers indicate where each area starts and ends. For example, in a generated index, entries that refer to multiple locations use different hypertext markers for each page number, making each page number for the entry a different active area.

When you convert a FrameMaker version 3 hypertext document to version 4, if there are any paragraphs that include more than one hypertext marker, the links might not work or they might not highlight correctly. Look for incorrect hypertext markers and delete them.

For more information about active areas in text, see “Hypertext and view-only documents” in your Frame user’s manual.

- The message `system` command in a hypertext marker

If you use a relative path in this command, the path is relative to `$FMHOME`, not to the location of the document that contains the hypertext marker. This is not new to hypertext, but it was not documented in earlier versions.

However, you can use `$FMDOCDIR` to specify the absolute path for the current document and `$FMDOCNAME` to specify the name of the current document file.

---

# 1

## ***Using FrameViewer on Multiple Platforms***

---

### **Section 1-1**

### **Cross-Platform Issues<sup>1</sup>**

---

#### **Pathname Conventions**

The only cross-platform hypertext links that are fully supported in FrameMaker 4 are those that use Unix style naming (i.e. “/”) and where NO SPACES are allowed in any component of the filename.

The above rules are fully enforced on the Macintosh version of FrameMaker 4 when you have Unix compatibility chosen in the Preferences dialog. If you have the Macintosh compatibility chosen, then slashes are still used, but you can have spaces in the filenames. These types of hypertext links are not cross-platform compatible, and this is a convenience factor for Macintosh-only online documents. In addition, the Macintosh product does not support colon (‘:’) based filename structures in hypertext links (meaning the paths to documents should not use colons in the path names). It conflicts with hypertext markers and other specific commands, such as “firstpage”.

The DOS-style pathnames (A:\mydir\mydoc) in hypertext links are NOT cross-platform compatible and are discouraged.

#### **Graphic Formats**

TIFF file format is consistent across all hardware platforms and it is the easiest choice for images. However, color TIFF images may not be a good choice because they display very slowly on some systems (i.e., 386-based systems with no RAM dedicated to the video). Convert color images to monochrome if possible to avoid color display issues. Line art maybe a better alternative since it is cleaner and easier to read. Line art also avoids all the potential image display issues.

---

1. See the *Changing Setup Files* document under online Manuals in the FrameMaker 4 Help for additional information.

---

## Section 1-2 Context-Sensitive Help

---

FrameMaker 4 allows you to develop online documents on any FrameMaker 4 supported platform, however, to develop context-sensitive help the development methods vary by platform. The following platform-specific sections will provide additional information for developing context-sensitive applications using FrameViewer 4 under Unix, Windows, and Macintosh.

---

## Section 1-3 Unix Specific FrameViewer Information

---

### Entering Command-Line Options for FrameViewer 4 for Unix

To start FrameViewer with command-line options, type the command to start the program, a space, an option name, a space, and a value if needed. Begin each additional option with a space. For example, the following command starts FrameViewer and opens a file:

```
viewer -file filename or -f filename
```

### Specifying a File to Open on Start-Up

Command-line options allow you to specify a file to open automatically when FrameViewer starts.

Start FrameViewer with this Option	To
<code>-file filename</code> or <code>-f filename</code>	Open the file and display its document window
<code>-fileIconic filename</code> or <code>-fi filename</code>	Open the document and display it as an icon
<code>+api</code>	Start API client programs when FrameViewer starts. See “API clients” on page 24 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-api</code> <code>-noapi</code>	Start FrameViewer without starting API client programs. See “API clients” on page 24 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-apiclient</code>	Start a specially designed API client program in a batch mode. See “API clients” on page 24 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.



Start FrameViewer with this Option	To
<code>-clientBitmapSize n</code>	Specify the size (in bytes) of the memory buffer that holds the screen representation of imported images and font information on the client running FrameViewer. See “Memory allocation for images and fonts” on page 18 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-geometry wxh+x+y</code>	Specify the size and coordinates of the main Frame Viewer window. For more information about specifying coordinates, see “Window positions and dimensions” on page 21 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-maxClientBitmaps num</code>	Specify the maximum number of images (both bitmaps and EPSI files) that the client running the FrameViewer product can keep in memory. See “Memory allocation for images and fonts” on page 18 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-maxServerBitmaps num</code>	Specify the maximum number of images (both bitmaps and EPSI files) that the server displaying the FrameViewer product can keep in memory. See “Memory allocation for images and fonts” on page 18 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-noViewerFile</code>	Start FrameViewer without prompting the user to open a file. See “Resources for controlling FrameViewer” on page 28 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>-serverBitmapSize n</code>	Specify the size (in bytes) of the memory buffer that holds the screen representation of imported images and font information on the server displaying the Frame Overproduce. See “Memory allocation for images and fonts” on page 18 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
<code>+splashscreen</code>	Display the licensing window when starting a FrameViewer product. See “Start-up messages” on page 24 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.

Start FrameViewer with this Option	To
-splashscreen -nosplash	Do not display the licensing window when starting a FrameViewer product. See “Start-up messages” on page 24 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
+viewerIsServer <sup>1</sup> -server	Start FrameViewer without displaying the main window. See “Resources for controlling FrameViewer” on page 28 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.
-viewerIsServer	Start FrameViewer and display the main window. See “Resources for controlling FrameViewer” on page 28 of the <i>Changing Setup Files</i> document. The <i>Changing Setup Files</i> document is located under online Manuals in the FrameMaker 4 Help.

1. Note that if you suppress the viewerkit, there is no mechanism to exit FrameViewer. This can only be done through the GUI EXIT button, or through external RPC client communication. The only other way to kill the Viewer session is through an external program/script that finds the PID under of the session and issues and O/S kill signal.

## Minimizing the Unix FrameViewer Installation

1. Which files can be safely removed from a Unix FrameViewer installation tree so as to make the smallest version needed to run the FrameViewer product only?  
Request Frame Technical Note #982 from Frame’s Technical Support at comments@frame.com or M — F, 6 — 5 PST at (408) 922-2744.
2. If I have limited the fonts used within the online documents to Times, Courier, Helvetica, and Symbol in bold, underline, and italic faces. Can I reduce the number of supporting fonts in the FrameViewer Unix Directory?  
The fonts in the file \$FMHOME/fmunit/fontdir/fontlist are the ones available within the product. Any font not used in your online document set can be deleted. See *Managing Frame Products* about “Installing Fonts” and reverse the procedure of editing the fontlist for any files not needed. The corresponding .afm and .bfont files can be removed.  
All the dialog boxes’ fonts are supplied by the Xserver and are not part of the FrameMaker/Viewer tree.
3. Can the different FrameViewer binaries for Unix use a single font set?  
The entire FrameViewer for Unix product line at equivalent revision numbers (e.g., 3.1a or 4.0.3) are explicitly designed to install into the same directory tree. Only the binary programs (the viewer executable, the print driver, filters, utilities) which are be different for various architectures, are redundant, however, all other files are common to all platforms. See “Installation Notes — Installing Frame Products on Multiple Platforms”.

4. How can we control FrameViewer window placement in Unix?

Window placement is controlled through xresources. Resources for dialogs can be found in `$FMHOME/fm_init/langdir/xresources/Maker` for FrameMaker version 4. If you are delivering FrameViewer on a variety of platforms, use a conservative number towards the upper-left corner of the X root window. This prevents users of low-resolution and/or small screens from having dialog boxes appear off their screens.

The documents window's coordinates are stored when the document is saved, and cannot be specified interactively or by external resources.

5. Can FrameViewer be modified not to display its opening window and only start our opening document?

(See "Entering Command-Line Options for FrameViewer 4 for Unix" on page 22.)

6. Can PATH variables be set to avoid placing specific paths to files within hypertext markers?

We do not recommend the specific shell variable "`$PATH`", the marker windows will use system environment variables in their path descriptions. This is not a search path, but a specific tree substitution.

You can use the environment variables `$FMDOCDIR` and `$FMDOCNAME` with the message system commands. When the viewer clicks an active area containing the command, `$FMDOCDIR` is set to the directory containing the current document and `$FMDOCNAME` is set to the name for the current document.

For example, the command `message system cp /HyperDir/Data $FMDOCDIR`

copies the file named Data from the HyperDir directory to the directory containing the current document.

The command `message system cp /HyperDir/$FMDOCNAME.Data $FMDOCDIR`

copies the file named the same as the current document with the extension Data from the HyperDir directory to the directory containing the current document.

---

## Licensing

1. Does the floating license scheme for FrameViewer operate in the same fashion as it does for Frame Installations?

You would need to negotiate an embedded licenses agreement for OEM to end-user distribution with FrameViewer, in which case the Floating License Server may or may not be necessary. Depending on the agreement, you may decide to supply instructions to your end-users on how to set-up a Floating License Server in conjunction with your product.

2. What environment variables must be set for each user to access the floating license on FrameViewer Unix?

There are two licensing variables required for Version 3.1 and two for version 4.0. `FRAMEUSERSD_HOST` (v3) to identify the Floating License `FM_FLS_HOST` (v4) servers host machine.

`FRAMEUSERSD_AUTO` (v3) to identify the start-up script to be `FM_FLS_AUTO` (v4) executed on the host matching if the client viewer finds it is not running.

## Context-Sensitive Help for Unix

To develop context-sensitive help in another application using FrameViewer under Unix, the following functionality is available: in the source bin `$FMHOME/bin` there is an `fmclient` program that can send messages to FrameViewer. The `fmclient` program allows you to open a doc and jump to a specific hypertext marker as well as connect to an already running FrameViewer process. To find out more about `fmclient`, type `fmclient -h`.

The `fmclient` program is the same as `clclient`. In version 3, the source code for `clclient` was in the install directory, `$FMHOME/source/openmaker/clclient`. This code will also work with FrameMaker 4.

In version 4, `clclient` (the uncompiled version of `fmclient`) is included as part of the Frame Developer's Kit (FDK), and is in the `$FMHOME/fdk/apps/unix.apps/frameserver/clclient` directory. If you do not have the FDK and you need the uncompiled version of `fmclient` it is available to you via ftp:

Frame FTP: `ftp.frame.com -or- 192.111.118.8`

Login as anonymous. Type `LS` to see files and directories.

Read-me files contain directions and file trees in each Directory.

When you login or `CD`, you will get a message file with directions and usage. Files `*from*` Frame are in `/OUTGOING`, unsecure files `*for*` Frame Technical Support are in `/TECHSUP`. Secure files for Technical Support go in the `/PRIVATE` directory. `/PUB` is a directory for people that are not working with Technical Support. Please do not leave files in the `ROOT` directory.

If you need the `clclient` model of connecting to an already running FrameViewer process, but also need to access the API functionality of the FDK, you may want to develop an application that combines `clclient` and the FDK API. For instance, you may want to add menu items to the FrameViewer or perform processing when a notification such as file open or close is received.

**Example Scenario.** Suppose in an application any time a user selects help from “application X” you want a FrameViewer document called “main\_help.doc” to be displayed. In addition, suppose that you want to add various menu options to the FrameViewer help implementation.

This is how you would achieve this functionality:

- Use `clclient` to bring up the initial FrameViewer document “main\_help.doc”. That is, “application X” would contain code to make the call “`clclient -f main_help.doc`”. The `clclient` program would check to see if FrameViewer is running; if FrameViewer is not running, it will be started, if FrameViewer is running, the running FrameViewer will be sent a message telling it to open “main\_help.doc”.
- When FrameViewer is brought up the first time, set it up so it runs `FV_help_apiclient`, which adds the various menu options.

In other words, you can take advantage of `clclient`'s ability to accept asynchronous messages, to determine if FrameViewer is already running and to raise/open the “main\_help.doc” window. When `clclient` needs to start FrameViewer, it also starts the `FV_help_apiclient`. The `FV_help_apiclient` adds the menu options, keeps track of what help files are open and provides you with a more robust integration.

**Command Line Options for fmclient.** Use the -h (or -help) option:

```
$FMHOME/bin/fmclient -h
```

```
fmclient [-display host:dpy] [-rpcProp propName] [-fpname  
command]
```

```
fmclient [options...] -n filename
```

```
fmclient [options...] -p filename
```

```
fmclient [options...] -f filename [linkname] [Y|N]
```

-fpname is to specify the command to start the Frame product, (e.g. \$FMHOME/bin/imaker -deutsch)

-n opens a template as a new file

-p prints a file to \$FMPRINTER (\$FMPRINTER must be defined)

-f opens an existing file, "linkname" specifies a hypertext link to jump to. If the file specified is already open, Y tells it to open a new window, N not to. The default is N. If the file specified is a brand new file (not already open), Y|N has no affect (i.e., the file will always open in a new window).

If you wish to open a program other than 'maker', set the environment variable:

\$FM\_PROGNAME to be viewer or imaker or builder, etc. Alternatively, the command-line flag, "-fpname" stands for "Frame Product Name" and can have the same arguments as the environment variable.

To open a FrameViewer file without the Open dialog box appearing, set the following environment variable to read by fmclient:

```
FM_PROGNAME="viewer -noViewerFile"
```

or use the X resource:

```
Maker.noViewerFile: True
```

---

## Section 1-4

## Windows Specific FrameViewer Information

---

### Context-Sensitive Help for Windows

To develop a context-sensitive help system under Windows you will need the Windows version of the Frame Developer's Kit (FDK). The Windows version of the FDK allows other applications to asynchronously control the Frame product through an FDK client.

**How is Asynchronous Communication Initiated?** Asynchronous communication is initiated by the application that wants to control the Frame session. This application sends a message to the Frame product via the Windows function `SendMessage()` in which it specifies the FDK client and the command (an integer) it wants the client to carry out. For example,

```
SendMessage(frame_hWnd, (WM_USER+12), calling_app_hWnd,  
            MAKELONG(fdk_dll_instance, (intCommand)));
```

After the application's call to `SendMessage()` has returned it can continue carrying out its own processing.

The Frame product must be running in order for your application to send it a message. So, the application should check to see if Frame is running and if it is not then it can start a Frame session by calling the Windows function `WinExec()`.

**How Does Frame Respond?** The Frame product posts a deferred message to itself and then returns one of the values listed below:

- 99 if the Frame product is busy and can't process the request
- 1 if the ID of the FDK client is bad or some other error occurred
- zero if the Frame product has successfully noted the request.

The Frame product responds to the deferred message by calling the FDK client's `F_ApiCommand(command)` callback function where `command` is the `(intCommand)` that was specified by the application in the `SendMessage()` function call. The FDK client then calls FDK functions to perform document manipulation in the Frame session.

When the Frame product has carried out all of the FDK functions called by the FDK client it notifies the other application that it and the FDK client have finished processing its request. The Frame product notifies the application of this by calling the Windows function `PostMessage()`.

The application must have executed the Windows function call `RegisterWindowMessage("FM_ApiNotify")` in order to receive the message posted by the Frame product.

---

## Section 1-5

## Macintosh Specific FrameViewer Information

---

FrameViewer on the Macintosh supports the four following required AppleEvents:

- Open Application
- Open Document
- Print Document
- Quit Application

Knowing this, you can use a utility such as AppleScript to send AppleEvents from within your application.

Another method you can use is to develop an api client with the Frame Developer's Kit (FDK) for the Macintosh. This would involve coding AppleEvent handlers in the api client for the AppleEvents you wish to send/receive. This method allows you to develop your own class of events that do exactly what you want within the Frame product. Using this method, you send the AppleEvents to the Frame product and because your api client has registered its handlers with the Frame session it appears as if Frame supports the AppleEvents you need.

---

# 1

## *Distribution/CD-ROM Production*

---

### Section 1-1

### CD-ROM Technology

---

#### **Why Manufacture a CD-ROM?**

When CD-ROM technology was first introduced, many people considered CD-ROM production a mysterious, expensive, and time-consuming procedure. Today, companies can produce a simple CD-ROM quickly and at a reasonable cost.

The vast storage capacity of CD-ROM, the low cost of CD-ROM drives, and the standards that are in place for the media, all combine to make CD-ROM the ideal way to distribute large amounts of information at a low cost. CD-ROM also is a cost effective way to distribute, publish, or gain access to information that you or your company works with on a regular basis.

#### **Types of Data Stored on CD-ROM**

The general rule to keep in mind when considering CD-ROM is: data is data. The only differences between data stored on a hard disk and data stored on a CD-ROM disc are the manner in which the data is arranged and the fact that, after committed to CD-ROM, the data cannot be changed. The data must be in a static form, which means that it needs no further changes. You can make changes only by updating the data and producing a new disc.

#### **Updating CD-ROM Discs**

In today's world, information is rarely static. Frequent updates make useful information more expensive to acquire and produce on disc. Yet some of the most useful and powerful CD-ROM applications are those that are updated regularly.

Update frequency is determined by the nature of the data. Information that changes regularly, however, may be suited for CD-ROM storage.

If the information a customer wants to store is prone to change, they must evaluate their willingness to update the CD-ROM. Updating the disc means future costs for a new CD-ROM that contains the new information.

#### **How Much Does It Cost?**

The cost of producing a CD-ROM title depends on many factors. The actual cost of physically replicating a CD-ROM disc by mass production (500 or more copies) is about \$1.50 per disc.

Previously, a small stamping of discs (one to 25) was rather expensive. With the introduction of one-off CD-ROM writers from companies such as Sony, JVC, Ricoh, and Philips, the cost of producing a small quantity of CD-ROMs has decreased. You can for example purchase your own desktop disc mastering system for around \$3,800. If you take the data through the production process, you can have a single CD-ROM cut by a service bureau for as little as \$99. Mastering facilities also offer special rates (near \$500) for one to ten prototype discs.

The demand for affordable CD-ROM publishing prompted the CD-ROM industry to find better and cheaper ways to make CD-ROM available to a wider market

Because consumers can farm out parts of the process to service bureaus, they no longer need thousands of dollars equipment. Today, small businesses and individuals can produce CD-ROMs on their desktops.

**CD-ROM Development  
and Production  
Processes**

<b>Process</b>	<b>Purpose</b>
Data acquisition	Assembling the information to be placed on the CD-ROM
Software acquisition	Choosing an indexing and retrieval program that is suitable for your data
Data Conversion	Altering the data to a format that can be handled by the indexing and retrieval software
Indexing	Using the indexing and authoring software to catalog the data for searching and linking audio and graphics
Premastering	Putting the data in CD-ROM format while it still resides on a hard disk
Testing	Simulating the CD-ROM environment so that the application's operation can be tested and optimized before committing it to a final premastered image to send to a mastering facility
Mastering	Producing the glass master and steel stamper for replicating CD-ROM discs



**CD-ROM Manufacturing Facilities**

The following is a list of just some of the manufacturing (replication) facilities in the United States and Canada.:

<p><b>Allied Records Corp.</b> 6110 Peachtree Street Los Angeles, CA 90040 213-725-6900</p>	<p><b>KAO</b> 41474 Christie Street Fremont, CA 94587510-657-8425</p>
<p><b>Alshire International, Inc.</b> 1015 Isabel Street Burbank, CA 91506 213-849-4671</p>	<p><b>KAO</b> 1857 Colonial Village Lane Lancaster, PA 17601 717-392-7840</p>
<p><b>Americ Disc Inc.</b> 2525 Canadien Drummondville, Que. Canada J2B 8A9 819-474-2119</p>	<p><b>Metatec/Discovery Systems</b> 7001 Colonial Village Lane Dublin, OH 43017 614-761-2000</p>
<p><b>American Multimedia Inc.</b> 2609 Tucker Street Burlington, NC 27215 910-229-5554</p>	<p><b>Nimbus</b> PO Box 7427 Charlottesville, VA 22906 804-985-1100</p>
<p><b>ASR Recording Services of CA</b> 8960 Eton Avenue Canoga Park, CA 91304 818-341-1124</p>	<p><b>Pilz Compact Disc.</b> PO Box 220 Concordville, PA 19331 215-459-5035</p>
<p><b>BQC, Inc.</b> 2101 South 35th Street Council Bluffs, IA 51501 712-328-8060</p>	<p><b>Polygram Mfg. Dist. Centers, Inc</b> PO Box 400 Grover, NC 28073 704-734-4100</p>
<p><b>Capitol Records</b> 1 Capitol Way Jacksonville, IL 62650 217-243-8661</p>	<p><b>Print NW/Six Sigma</b> 4101 Industry Drive East, Suite D Fife, WA 98424 206-922-0990</p>
<p><b>Cinram</b> 2255 Markham Road Scarborough, Ont. Canada M1B 2W3 416-298-8190</p>	<p><b>Sanyo Laser Products</b> 1767 Sheridan Street Richmond, IN 47374 317-935-7574</p>
<p><b>Cinram</b> 1600 Rich Road Richmond, IN 47374 800-927-7749</p>	<p><b>Sonopress USA</b> 108 Monticello Road Weaverville, NC 28787 704-658-2000</p>

<b>Denon</b> 1380 Monticello Road Madison, GA 30650 706-342-3425	<b>Sony DADC</b> 400 N. Woodbury Avenue Pitman, NJ 08071 609-589-8000
<b>Distronics</b> 2800 Summit Avenue Plano, TX 75074 214-881-8000	<b>Sony DADC</b> 3181 N. Fruitridge Street Terre Haute, IN 47804 812-462-8160
<b>DMI</b> 1120 Cosby Way Anaheim, CA 92806 714-630-6700	<b>Technidisc</b> 2250 Meijer Drive Troy, MI 48084 810-435-7439
<b>DMI</b> 4905 Moores Mill Road Huntsville, AL 35810 205-859-9042	<b>3M</b> 2933 Bayview Drive Fremont, CA 94538 510-440-8161
<b>Evatone</b> 4801 Ulmerton Road Clearwater, FL 34622 800-382-8663	<b>3M</b> 1425 Parkway Drive Menomonie, WI 54751 715-235-5567
<b>HRM/HMG</b> 15 Bilpin Avenue Hauppauge, NY 11788 516-234-0200	<b>Uni Distribution</b> Highway 154 East Pinkneyville, IL 62274 618-357-2167
<b>JVC</b> 2 JVC Road Tuscaloosa, AL 35405 800-677-5518	<b>US Optical Disc</b> 1 Eagle Drive Sanford, ME 04073 207-324-1124

**Section 1-2**

**Unix Distribution Issues**

When you publish documents in FrameViewer format and deliver them along with FrameViewer on CD-ROM there are some issues to consider.

There are some limitations to CD-ROM format, most notably the “8.3” file naming restriction. Frame delivers its Unix installation in tar volumes on an ISO/9660 CD-ROM for this reason. The tar volumes are 8.3, but the filenames within are not and need to be installed onto “long” filename systems (4.2, hfs, ufs, etc.). The filenames in the FrameMaker/Viewer directory tree are fixed and necessary for binary operation and cannot in any way be shortened for publishing in compliance with ISO9660. For this reason, you need to create a tar or some other image that could be extracted to an end-user’s filesystem for FrameViewer operation.

In addition, there are constraints even publishing cross-platform capable documentation on CD-ROM. Hewlett-Packard has implemented a version number for files on 9660 filesystem. Files that would be read, for example as "filename.doc" (8.3) on a Sun-mounted CD-ROM would be seen by the HP as "FILENAME.DOC;1". The implications of this can be seen in our manual "Installing FrameMaker for HP workstations". Just for installation, we tell the end-user to double quote the filename to protect the semicolon against shell interpretation, and to enter the file in all caps. The same CD-ROM mounted on a Sun has no such restrictions.

This has an impact within a hypertext document published on CD-ROM. One solution may be to create a separate tree, with new links created for each hypertext link, created for the HP system. All those filenames would be referenced in the links as "A\_FILE.DOC;1" rather than the "a\_file.doc" needed for the other systems. However, there is no mechanism in FrameMaker/Viewer to automatically resolve these differences. Note that if only one document tree is produced, the tree needs to be installed onto the end-user's hfs (HP) filesystem so as to avoid the "semicolon-1" problem.

---

## Section 1-3

## MS Windows Distribution Issues

---

### Uninstalled Application

The following section outlines some distribution methods that might be used to distribute documents under FrameViewer for MS Windows.

Copy (using the DOS command `xcopy` or something comparable) the original FrameViewer diskettes to the master media. The contents of each diskette needs to be in its own directory. For example, create a directory called `\disk1` for Disk 1. In addition, make a directory for the documents you are shipping with FrameViewer (these are the documents you want the end users to view).

For the next step, you need to write an installer. There are software packages available that can assist you with this procedure.

The installer will need to:

- verify there is enough disk space to install FrameViewer and the distributed documents
- invoke the FrameViewer installer (the end user will need to provide a registration number for FrameViewer)
- install your help documents

The FrameViewer installer will make the necessary entries in `win.ini` and create the program group and icon in MS Windows.

This is the preferred method if FrameViewer and the distributed documents are to be installed on a hard drive or network drive, as it allows FrameViewer to run the fastest.

### Installed Application Uncustomized

With this method, FrameViewer and all of the documents are read directly from the media instead of being "installed". To prepare this type of distribution, first install FrameViewer with all the desired options to a hard drive and make the desired changes within the `viewer.ini` file. (These will be the fixed settings for all users.)

Copy the installed FrameViewer or “image” along with the distribution documents to the master media. If you purchased FrameViewer Professional, `viewer.exe` is shipped with the product (`viewer.exe` is not included with FrameViewer). You can download `viewer.exe` from Frame's Bulletin Board at (408) 975-6729 or (408) 975-6730. This file allows FrameViewer to be launched by clicking on the filename `viewer.exe` or a file associated with `viewer.exe` in the Windows File Manager.

With this method, there is not a FrameViewer icon or icons for your documents to start-up FrameViewer. You must use Windows File Manager, or `FILE/RUN` in Program Manager to start the application.

It is not necessary to write an installer for this type of installation; however, you must make the following entries within the `win.ini` file:

```
[FrameViewer4]
FMHome=x:\viewer4
IniFile=x:\viewer4\viewer.ini
```

Where `x:` is the drive letter of the distributed media, `\viewer4` is the absolute path to the FrameViewer directory and `\viewer4\viewer.ini` is the absolute path to the `viewer.ini` file.

This method is suited for end users needing to launch FrameViewer from the distributed media. If your distributed media is CD-ROM, FrameViewer will run slower than if it were installed locally or on a network drive. CD-ROM is one of the slowest media options and this is true for any MS Windows application.

## Installed Application Customized

This is more complicated to setup, but is much more customizable for the end user. Like the previous method, you will copy an image of the installed FrameViewer application to the master media along with the documents for distribution. You will not need to edit the `viewer.ini` file since this will not be the same for all end users.

You will need to write an installer to do the following:

- Ask the end user for a drive to copy a minimal set of files to
- Check the destination drive to verify enough space exists (approximately 3.2MB)
- Copy the files in the `\viewer4` directory to the destination drive including `viewer.ini` and `frame.exe`. It is not necessary to copy the subdirectories of `\viewer4`. (This may be a little different if you are using Viewer Pro with Bookmarks and Notes.)
- Edit `win.ini` and add the following information:

```
[FrameViewer4]
FMHome=x:\dirname
IniFile=x:\dirname\viewer.ini
```

Where `x:` is the drive letter for the FrameViewer installation, `\dirname` is the absolute path to destination drive for FrameViewer and `\dirname\viewer.ini` is the absolute path to the `viewer.ini` file.

- Edit the copy of `viewer.ini` in the installer destination directory so it's referring to the distributed media. This way only the `.exe` and `.ini` files will need to reside on the local hard drive, the rest of the files remain on the distributed media, which will aid system performance.

- Add a windows group and icon. The icons may point to your documents as follows:

```
Name:Widget Specification Report
Path:q:\dirname\FRAME.EXE -VIEWER x:\ROOT\subdir\WID-
GET.RPT
Starting directory: q:\dirname
Icon: z:\ICONS\CUSTOM.ICO
```

Where x: is the drive letter of the distributed media, q: is the drive letter where the copy of \VIEWER4 is located, and z: is the drive letter where the icons reside which you made to represent the documents.

You can create the icons and the other information ahead of time and have the installer just place them into the group.

The “Installed Application Customized” is the best way to setup FrameViewer on a CD-ROM. The end user can have an icon for each document or book file. This technique provides the best performance while providing customized options during installation. In addition, each end user's setup can be unique. Note, however, an entry must always be made within the `win.ini` file for FrameViewer to launch.

If you are using Notes and Bookmarks of FrameViewer Professional, your installer may also have to write the correct entries within `nts.cfg`.

You must purchase the right to distribute FrameViewer with your documents whether on CD-ROM or any media type. If you are interested in doing so, contact any PC software reseller that carries FrameMaker or Frame Sales at (408) 975-6000. In Europe, please contact your local Frame Reseller or Frame Europe on +(44) 081 606 4100.

Blank

---

# 1

## Troubleshooting

---

**ISSUE 1:** We have FrameViewer for Unix as a help engine provided by another company. In addition, we also have FrameMaker for Unix internally. When we start FrameViewer, we get an error that there are no licenses installed. Some users also get an error about not being authorized to use the product when starting FrameViewer or FrameMaker, while others are able to use both concurrently.

How can we configure these two products to co-exist?

**ANSWER:** The problem occurs because two license servers are needed; one for FrameMaker and one for the embedded FrameViewer. For FrameMaker, all licenses can be administered by a central server. Identification of the server in the user's environment is controlled by two system variables:

1. FM\_FLS\_HOST= hostname
2. FM\_FLS\_AUTO=/license/server/startup/script

FrameMaker/Viewer pings the FM\_FLS\_HOST for a live server and if it does not receive a response, it executes the script named in \$FM\_FLS\_AUTO. For FrameViewer embedded licenses, the value of the FM\_FLS\_HOST should be 'localhost' and the auto-start script should be in \$HOME/fm\_init. This conflict presents a problem for those running both Maker and embedded Viewer licenses.

The simplest workaround for the system administrator is to wrap the maker and viewer commands in a shell script which first sets the FM\_FLS\_\* variables to the correct value for the product, and then starts the appropriate product.

For example:

```
#!/bin/sh
# shell script startup wrapper for Viewer and Maker
# This should be in the path before $FMHOME/bin and
# $FMHOME
# must be defined. This sample has not been tested and is
```

---

```

# provided as a point of departure for the interested
# system administrator.

usage () {
    echo "Usage: maker | viewer [arg ...]
}

if ( "$#" -eq "0" ); then
    usage
    exit 1
else
    myname=$1
    shift
    case $myname in
        maker | imaker | builder | ibuilder )
            . $FMHOME/fm_init/env.sh
            $FMHOME/bin/$myname "$@"
            viewer )
                FM_FLS_HOST=localhost
                export localhost
                . $FMHOME/bin/viewer "$@"
            *) #error case
                usage
                exit 1

    fi
#
# End example

```

If a FrameMaker/Viewer user runs on the license server (e.g., an X-terminal), then that user must have a reserved license on another host, as only one `fm_fls` process can run on any one machine.



---

**ISSUE 2.** When I automatically generate hypertext links in a TOC, Index, etc., my hypertext markers are not active.

**ANSWER.** Make sure that your filenames are greater than one character.

---

**ISSUE 3:** When I access a hypertext link to page 2 of a document, FrameViewer displays 2 pages (half of page 1 and half of page 2).

**ANSWER.** The problem is caused by the hypertext link being too close to the edge of the page. By moving the link farther away from the edge of the page, the problem is resolved.

---

**ISSUE 4.** When I generate a book or list I get Internal Error 148E08.

**ANSWER:** This error occurs when a FrameMaker Character Format has been applied to the `openObjectId` hypertext command on the Reference Page of the generated list document. Removing the Character Format on the Reference Page will eliminate the Internal Error.

---

**ISSUE 5.** I need my documents to support relative pathnames not absolute pathnames on the Windows platform. How can I do this?

**ANSWER.** In the `viewer.ini` file (located in the viewer home directory) add the following line in the `[Directories]` section:

```
[Directories]
```

```
Drive=c:
```

In any hypertext link in a FrameMaker document, use the following syntax:

```
gotolink <$path[drive]>/path/path/filename:firstpage
```

The drive letter “c:” will be picked up in the path so that the command will be evaluated to:

```
gotolink c:/path/path/filename:firstpage
```

If the documents are stored on any other drive, all you have to do is change the drive variable in the `viewer.ini` file.

