

© 2002 Adobe Systems Incorporated. All rights reserved.

Adobe FrameMaker 7.0 FDK Sample Clients

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, Acrobat Reader, Adobe Type Manager, ATM, Display PostScript, Distiller, Exchange, Frame, FrameMaker, InstantView, and PostScript are trademarks of Adobe Systems Incorporated.

Apple, PowerBook, QuickTime, Macintosh and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. Focoltone is a registered trademark of Gordon Phillips Limited. ImageStream Graphics Filters and ImageStream are registered trademarks of Inso Corporation. Microsoft, MS-DOS, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Quadralay and WebWorks are registered trademarks of Quadralay Corporation. PANTONE®, PANTONE MATCHING SYSTEM®, PANTONE Process Color System®, and POCE™ are trademarks of Pantone, Inc. Proximity and Linguibase are registered trademarks of Proximity Technology Inc. A Merriam-Webster is a registered trademark of Merriam-Webster Inc. Sun is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. TRUMATCH is a registered trademark of Trumatch Inc. Unix is a registered trademark and X Window System is a trademark of The Open Group. Verity and TOPIC are registered trademarks of Verity, Inc. All other trademarks are property of their respective owners.

The following are copyrights of their respective companies or organizations: Portions reproduced with the permission of Apple Computer, Inc. © 1996 Apple Computer, Inc. Milo © 1988 Ron Avitzur PANTONE® Computer Video simulations displayed may not match PANTONE-identified solid color standards. Use current PANTONE Color Reference Manuals for accurate color. "PANTONE Open Color Environment™ (POCE™)" © Pantone, Inc., 1994, 1996. Pantone, Inc. is the copyright owner of "PANTONE Open Color Environment™ (POCE™)" and Software which are licensed to Adobe to distribute for use only in combination with the Adobe Software. "PANTONE Open Color Environment™ (POCE™)" and Software shall not be copied onto another diskette or into memory unless as part of the execution of the Adobe Software. The Spelling and Thesaurus portions of this product are based on Proximity Linguistic Technology. The Proximity/Merriam-Webster Linguibase © 1983, 1990 Merriam-Webster, Inc. C.A. Stromberg AB; Espasa-Calpe; Hachette; IDE/AS; Kruger; Lluís de Yzaguirre i Maura; Merriam-Webster Inc.; Munksgaard Int. Publishers Ltd.; Nathan; Text & Satz Datentechnik; Van Dale Lexicographie bv; William Collins Sons & Co. Ltd.; Zanichelli. All rights reserved. Color Database © Dainippon Ink & Chemicals, Inc., licensed to Adobe. Outside In® Viewer Technology, 1992-1996 Inso Corporation; Image Stream® Graphics and Presentation Filters, 1992-1996 Inso Corporation. All rights reserved. TRUMATCH 4-Color Selector © 1992 Trumatch, Inc. All rights reserved. Portions copyrighted for the FrameViewer Retrieval Tools © 1988-1995 Verity, Inc. All rights reserved.

APPLE COMPUTER, INC. ("APPLE") MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE. APPLE DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE APPLE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE APPLE SOFTWARE IS ASSUMED BY YOU. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU.

IN NO EVENT WILL APPLE, ITS DIRECTORS, OFFICERS, EMPLOYEES, OR AGENTS BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE OR INABILITY TO USE THE APPLE SOFTWARE EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA

Notice to U.S. government end users. The software and documentation are "commercial items," as that term is defined at 48 C.F.R. §2.101, consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. government end users (A) only as commercial items and (B) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in the Adobe standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Sample	Description	Win	Unix	Mac
abc	Demonstrates how to create and maintain a client owned inset. It works in combination with the <code>geoserver</code> sample client. Choose menu items provided with this client to insert the text insets, then use the additional menu choices from the <code>geoserver</code> client to make modifications to those text insets.	✓	✓	✓
addstructrow	Demonstrates how to add a row to an existing structured table.	✓	✓	✓
aframes	Generates a document that lists the page number, position, and size of all the anchored frames on body pages in the active document.	✓	✓	✓
async	Demonstrates asynchronous clients on Unix. This client can be compiled into two different sample clients to show different aspects of communication with a FrameMaker process. In one version, the sample client adds a menu to the document window menu bar and then waits for the end user to choose an item from the menu. In the second version, the sample client forcibly adds text to an open document. The client is described in detail in the Unix FDK Platform Guide. NOTE: To compile this client, you must make some edits to the Makefile. The required changes are detailed in comments within the Makefile.		✓	
badlinks	Generates a document that lists unresolved hypertext links. This document lists the page number and marker text of any hypertext marker with an unresolved link to a document, page, cross-reference, or graphic object. The client also writes errors to the console.	✓	✓	✓
clickprt	Brings up the print user interface for the filename that is passed as an argument in the hypertext message. Responds to a “message client”-style hypertext command.	✓	✓	
cond	Adds or removes the comment condition for the currently selected text.	✓	✓	✓
ddeclnt	Demonstrate registering FrameMaker client as a DDE server. This client registers as a DDE application the client named <code>FrameDdeSample</code> . The client accepts both <code>execute</code> and <code>request</code> transactions.	✓		
dealloc	Constructs and then displays a message in an alert box. The client builds a list of paragraph formats in the current document, determines the size of the list, then displays the number of paragraph formats in an alert box. This client demonstrates how to free resources allocated during string function calls.	✓	✓	✓
dialog	Creates a sample dialog box. A <code>.dre</code> file for the dialog box is included in the source. See the appropriate FDK Platform Guide for information about creating and compiling a dialog box resource from this file.	✓	✓	✓
eewriter	Converts upper case characters to lower case as the user types in a document.	✓	✓	✓

Sample	Description	Win	Unix	Mac
elemutil	Demonstrates how to manipulate elements and other FrameMaker structure objects. This client adds a menu called Element-Utils with the following commands: <ul style="list-style-type: none"> ● Format — Inserts the Emphasis element at the insertion point, or wraps the current selection in Emphasis. It then sets the Type attribute. ● Collapse All — Collapses the entire structure view. ● Expand All — Expands the entire structure view. Collapsing and expanding demonstrates how to traverse the structure tree. ● Reset Attributes — Resets all attributes to their default values. ● Add Acronym — Adds an Acronym and Defn element pair alphabetically to the AcronymList element. Demonstrates use of a custom dialog box and how to add an element. Source includes a .dre file for the dialog box. See the appropriate FDK Platform Guide for information about creating and compiling a dialog box resource from this file. 	✓	✓	✓
filter	The sample filter client for the filter that is used in examples throughout the FDK documentation. This client filters text files into Frame product documents.	✓	✓	✓
font families	Displays a list of fonts that are available in the paragraph designer.	✓	✓	✓
geoserver	Works in conjunction with the abc client to demonstrate how to create and maintain a client owned inset.	✓	✓	✓
getset	Demonstrates how to get and set document and session properties. This client includes the following commands: <ul style="list-style-type: none"> ● BORDERS_CMD — Toggles Borders on and off ● ZOOM_CMD — Sets a document's zoom factor to values between 25% & 1600% ● HOST_CMD — Queries the current session for the current host name. 	✓	✓	✓
hello	Greets the user at product startup time.	✓	✓	✓
hello2	Provides a menu selection to display the alert messages “Hello, World!” or “Goodbye, World!”.	✓	✓	✓
hellowho	Greets the user with a custom greeting established at product startup time.	✓	✓	✓
hibye	Demonstrates how to use notifications. Upon receipt of an FA_Note_PostOpenDoc notification, the client calls the function sayHello() to display an alert. Upon reception of an FA_Note_PreQuitDoc notification, the client calls the function sayGoodbye() to display a different alert.	✓	✓	✓
import	Demonstrates how to import graphic files.	✓	✓	✓

Sample	Description	Win	Unix	Mac
includelists	<p>Uses MIF statements or book component properties to determine which book components to print, or into which components to import formats or element definitions. This creates an effect similar to the user interface for these actions in versions of FrameMaker prior to FrameMaker 6.0. The client uses MIF statements or book component properties to determine its behavior as follows:</p> <ul style="list-style-type: none"> ● File>Print Files in Include List — Prints every component in a book whose DefaultPrint MIF statement is Yes (or whose FP_PrintInclude property is set). ● File>Import>Formats to Files in Include List — Imports formats into the book components whose DefaultApply MIF statement is Yes (or whose FP_ImportFmtInclude property is set). ● File>Import>Element Definitions to Files in Include List — Imports element definitions into the book components whose DefaultApply MIF statement is Yes (or whose FP_ImportFmtInclude property is set). 	✓	✓	✓
mainloop	<p>This directory contains sample clients that illustrate how to program asynchronous clients that run in main loops.</p> <p>NOTE: To compile these clients, you must make some edits to the Makefiles. The required changes are detailed in comments within each Makefile.</p> <p>The sample clients in this directory are:</p> <ul style="list-style-type: none"> ● <code>doselect</code> A client that executes its own <code>select()</code> calls. The input file descriptor for this client is the the shell window from which you start FrameMaker. The client uses <code>F_ApiAlert()</code> statements to echo shell input in alert boxes. ● <code>noselect</code> A client that does <i>not</i> execute its own <code>select()</code> calls. The input file descriptor for this client is the the shell window from which you start FrameMaker. The client uses <code>F_ApiAlert()</code> statements to echo shell input in alert boxes. ● <code>xfmtest</code> A sample X Window System application. This client displays a modeless window with a menu. When the user chooses Quit from the menu, the client closes the active FrameMaker document (without saving changes) and exits. 		✓	
markers	<p>Demonstrates how to work with markers, and how to use menu separators. This client includes the following commands:</p> <ul style="list-style-type: none"> ● <code>INSERT_CMD</code> — Inserts a marker of a specified type. The user is prompted for the name of the marker type. If the user specifies a string for a marker type, the client inserts a new marker of that type. The client also prompts the user for marker text. ● <code>COUNT_CMD</code> — Count all anchors within a selected text range. 	✓	✓	✓

Sample	Description	Win	Unix	Mac
mcount	Counts markers in the active document, and reports the number back to the user.	✓	✓	✓
mmlimport	A filter client for opening and importing mml files.	✓	✓	✓
myapi	The example application presented in Chapter 1 of the FDK Programmer's Guide. It adds a menu to the document menu bar, with commands to: <ul style="list-style-type: none"> ● Close the current document ● Set the fill pattern of a selected object to black ● Add a page at the beginning of the current document. 	✓	✓	✓
newfmt	Displays two dialog boxes to get user input, then uses the input to create a new paragraph format. The client then adds the new format to the document's paragraph catalog.	✓	✓	✓
openbook	Performs a scripted open of all files in a FrameMaker book.	✓	✓	✓
openxml.doc	Demonstrates how to open an XML document and import it into FrameMaker.	✓	✓	✓
otherapp	A simple client that connects to a FrameMaker client which is registered as a DDE server. The client displays a small window for the user to connect to the FrameMaker client and send either <code>execute</code> or <code>request</code> transactions. The client interprets <code>execute</code> transactions as hypertext commands. For example, you can execute the following hypertext command syntax: <code>gotopage 3</code> . The client interprets specific <code>request</code> transactions. For example, a request of <code>CharTag</code> directs the client to retrieve the character tag of the beginning of the selection in the active document.	✓		
PageNo2	Demonstrates a way to number pages in a document separately from the numbering properties that are set up for a book.	✓	✓	✓
pcount	Counts the pages in all components of a FrameMaker book.	✓	✓	✓
pcount2	Counts the pages in a FrameMaker document.	✓	✓	✓
pickfmts	Allows the user to choose a group of paragraph formats to delete from the active document. This client uses a custom dialog box.	✓	✓	✓
reverse	The Reverse Translator reverses the order of two elements upon translation from SGML to structured FrameMaker, and vice versa. Upon export, it takes a FrameMaker document which contains the element <code>First</code> followed by the element <code>Last</code> and saves it as an SGML document with <code>Last</code> followed by <code>First</code> . This client illustrates the use of <code>Sw_ScanElem()</code> to change the order of elements on export. NOTE: To compile this client on Unix platforms, you must make some edits to the Makefile. The required changes are detailed in comments within the Makefile.	✓	✓	✓

Sample	Description	Win	Unix	Mac
saveas	Saves a document as PDF or HTML, depending on the menu choice. This client also demonstrates how to set the hierarchical level of PDF bookmarks via the FDK.	✓	✓	✓
setpdfdocinfo	Demonstrates how to set object properties for Tagged PDF.	✓	✓	✓
setxmpinfo	Demonstrates how to set XMP info through the API	✓	✓	✓
sgmlplus	This is a C++ development environment for writing Structure Import/Export API clients. Note that <code>sgmlplus</code> is written on top of the Structure Import/Export API—it is not a distinct C++ API. For information about this environment, see the description in the included file, <code>sgmlplus.fm</code> . NOTE: To compile this client on Unix platforms, you must make some edits to the Makefile. The required changes are detailed in comments within the Makefile.	✓	✓	✓
struct	This sample client lays a basic foundation for all SGML or XML import/export clients, no matter how complex they become. NOTE: To compile this client on Unix platforms, you must make some edits to the Makefile. The required changes are detailed in comments within the Makefile.	✓	✓	✓
sumcols	Sums the columns in a table. This client sums only integer values. To obtain a number it looks for digits at the start of a table cell. The sum is placed in a footer row at the end of the table. Values in header rows are ignored.	✓	✓	✓
tables	Demonstrates how to traverse and manipulate tables, rows, cells, and cell text. This client includes the following commands: <ul style="list-style-type: none"> ● ITEM1—Deletes text from the cell that contains the current selection. ● ITEM2—Gets the row and column position of a single selected cell. ● ITEM3—Gets the integer value representing the position of the last Body Row. ● ITEM4—Gets the Integer value representing the position of the first Body Row. ● ITEM5—Determines whether a cell is set to Straddled or not. ● ITEM6—Places the cursor at the end of the paragraph (within a cell) that it is currently inserted in. 	✓	✓	✓
tag	For any text range that has a format override of italic, this client changes that range to use the emphasis character tag.	✓	✓	✓
text	Demonstrates how to manipulate text. This client gets text from the document and prints it to console, adds text to the document, deletes text, changes text point size, and changes text color.	✓	✓	✓

Sample	Description	Win	Unix	Mac
uservar	Writes a user name into a variable named “User Name”, when an FA_Note_PostOpenDoc notification is received. User name is acquired via the session property FP_UserName.	✓	✓	✓
version	Gets and displays the major/minor version number of FrameMaker. Makes use of string functions. This client makes calls to the functions getProductName(), getVersionMajor(), and getVersionMinor(). For more information about these functions, see the included file cutils.c	✓	✓	✓
viewonly	Sets all open documents to view only, and adds some text to each document at its last insertion point location.	✓	✓	✓
whatpage	Identifies the Body page number of the selected object. This client demonstrates how to find the body page that contains the current selection for selected text or graphic objects.	✓	✓	✓
wordcnt	Counts words in the current document. It locates all flows on body pages in the document, and reads the content of every paragraph in each flow. It then counts the words in each paragraph. A character is defined to be either an “in-word” or a “not-in-word” character. A “word” is a group of one or more “in-word” characters—delimited on either side of the group by a “not-in-word” character, a paragraph break, or a hard (user-inserted) line break. In addition the program counts words in graphic text lines which appear on body pages of the document. Text in footnotes, tables, or generated from variables or cross references is counted. Neither text generated by autonumber strings nor text on master or reference pages (e.g. headers and footers) is counted.	✓	✓	✓