

FDK Release Notes



Adobe

VERSION 6.0



Frame Developer's Kit, 4/21/00

ADOBE SYSTEMS INCORPORATED
Corporate Headquarters
345 Park Avenue
San Jose, CA 95110-2704
(408) 536-6000

IMPORTANT NOTICE

© 2000 Adobe Systems Incorporated. All rights reserved.

Adobe FrameMaker 6.0 User Guide for Windows, Macintosh, and UNIX

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, FrameViewer, 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; Luis 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.

Contents

To go to a section, click on a topic below.

Chapter 1 *What's New in the FDK* 1

- Book window changes 1
- View-only books 2
- Book error log 2
- Numbering for documents and book components 2
 - Synchronizing component and document numbering 2
- Scripted book updates 3
- Book-wide command processing 3
- Miscellaneous book-related changes 4
 - F_ApiNewNamedObject() 4
 - F_ApiSimpleImportFormats() 4
- Changes to F_ApiFind() 4
- Changes to optimize PDF file size 4
 - Properties to identify named destinations 4
 - API client to optimize legacy documents 5
- Specifying Document Info for PDF 5
- Specifying values for Structured PDF 5
- New functions 5
- Modified objects 5
 - Book objects 5
 - Book component objects 6
 - Command objects 7
 - Document objects 7
 - Paragraph format objects 8
 - Session objects 9
 - Structural element objects 9

Chapter 2 *Samples in the FDK* 11

Windows® 11

- Macintosh 11
- SGMLPlus Development Environment 12
- Includelists 13
 - File>Print Files in Include List... 13
 - File>Import>Formats to Files in Include List... 13
 - File>Import>Element Definitions to Files in Include List... 13

Chapter 3 *Last Minute Information* 15

- Developing with the API on all platforms 15
 - Cannot import SGML fragments via F_ApiImport() without displaying dialog 15
 - F_ApiDeletePropByName() 16
 - FO_Arc object and objects representing Bezier curves 16
 - Traversing a list of invalid elements 16
- Developing with the API on Macintosh platforms 16
 - F_Printf() 16
- Developing with the API on Windows platforms 17
 - FA_Note_BackToUser notification 17
 - Labels with keyboard shortcuts 17
- Designing dialog boxes in Frame products 17
 - Editing DRE files in UNIX 17
 - Highlighting text in multiline text boxes 18
 - Relating image pop-up menus and radio buttons 18
 - Resetting a platform view in a DRE file 18
 - Testing a dialog box in Windows NT 18
- Updates to the FDK documentation 18
 - FDK Programmer's Guide and Reference 19

Contents

SGML API Programmer's Guide 19

1

What's New in the FDK

What's New for Version 6.0

With release 6.0, FrameMaker® products introduce the following, which impact the FDK:

- Changes to the book window
- View-only books
- Book error log that lists book-processing errors
- Changes to numbering for documents and book components
- FDK functions for scripted book updates
- Book-wide command processing
- Miscellaneous book-related changes
- Changes to `F_ApiFind()` that allow searching through all the components in a book
- Changes to optimize the size of PDF files you create when you save as PDF
- Changes to Save As PDF to include Document Information and structured PDF

Book window changes

The appearance and behavior of the book window has changed for version 6.0. Users can now determine whether the book window lists the file name of each component, or a snippet of the first text in each component document. You specify this setting in the `FO_Book` object via the `FP_TypeOfDisplayText` property. When displaying the text in a book component, the text to display is stored in the component's `FP_ComponentDisplayText` property. FrameMaker products modify the value of this property when the user adds a file to the book, updates the book, or saves the associated document.

The book window now displays an icon for the book file, as well as icons for each book component. You determine whether the book icon is selected via the `FP_BookIsSelected` property.

In earlier versions you could display text in the book window's status line by setting a value to the `FP_StatusLine` property. You can now use that property with `F_ApiGetString()` to get the text that last appeared in the status line.

View-only books

You can now save and open a book as a view-only file. Book objects now include view-only properties that correspond with those for document objects. Also, the parameters retrieved by `F_ApiGetOpenDefaultParams()` now include an `FS_OpenBookViewOnly` property, and the `FS_BookIsInUse` property now has an `FV_OpenViewOnly` value. With `F_ApiGetSaveDefaultParams()`, the `FV_SaveFmtViewOnly` value for `FS_FileType` now applies to books. Finally, `F_ApiOpen()` no longer returns the `FV_CantOpenBooksViewOnly` value for `FS_OpenStatus`.

Book error log

When updating a book, FrameMaker products now write errors to a file that lists each error and includes hypertext links to the location in the component document where the error occurs. For example, the error log lists unresolved cross-references, and includes hypertext links to them. The error log is implemented via an FDK client named `BookErrorLog`.

Your FDK client can post its own error messages, including hypertext links, by calling the `BookErrorLog` client. You call the client via `F_ApiCallClient()`, and pass a string to specify the content, formatting, and link for the error message. For more information, see the entry for `F_ApiCallClient()` in the *FDK Programmer's Reference*, or Chapter 14, "Frame Book Architecture" of the *FDK Programmer's Guide*.

Numbering for documents and book components

Version 6.0 includes new building blocks for volume and chapter numbers. Book and document objects include corresponding properties and values for these types of numbering. Volume and chapter numbering are intended to replace the old prefix and suffix properties for book components; the `FP_Prefix` and `FP_Suffix` properties have been removed for version 6.0.

The numbering method for each type of numbering in a document or component is specified via a corresponding `ComputeMethod` property. For example, `FP_PageNumComputeMethod` and `FP_VolNumComputeMethod` specify numbering methods for page numbering and volume numbering, respectively. Also, `FP_PageNumbering`, `FP_PgfNumbering`, and `FP_FnNumberingPerPage` are obsolete in version 6.0; use `FP_PageNumComputeMethod`, `FP_PgfNumComputeMethod`, and `FP_FnNumComputeMethod` instead.

Synchronizing component and document numbering

With the exception of `FV_ReadFromFile` (in book components), the numbering properties and values are now the same for both `FO_BookComponent` and `FO_Doc` objects. In this way, FrameMaker products can maintain the same numbering for components and documents. When you update a book, the FrameMaker product computes the numbering

based on the compute method. Then if necessary, it modifies the document or component so the numbering properties will match.

If the compute method in a book component is `FV_NumReadFromFile`, the product updates the component's numbering properties to match the document. Otherwise, it updates the document to match the book component. In this way, after updating a book the numbering properties and values for components and documents should always match. The only exception is that a book component may have compute methods of `FV_ReadFromFile`, but document objects don't use this method.

Scripted book updates

The FDK includes two new functions; `F_ApiUpdateBook()` and `F_ApiGetUpdateDefaultParams()`. These functions provide scripting to control how to update a book; they work much the same as the corresponding `F_ApiSave()` and `F_ApiOpen()` functions. For more information, see the entries for `F_ApiUpdateBook()` and `F_ApiGetUpdateDefaultParams()` in the *FDK Programmer's Reference*. You can also use `F_ApiCheckStatus()` to evaluate the parameters returned by `F_ApiUpdateBook()`.

`F_ApiPrintUpdateBookStatus()` prints errors returned by `F_ApiUpdateBook()`. It is useful for debugging your clients. For more information, see the entry for `F_ApiPrintUpdateBookStatus()` in the *FDK Programmer's Reference*.

Book-wide command processing

Users can now perform functions across selected components in a book. For example, users can search for a string in all the components of a book. For book-wide commands, the FrameMaker product posts an `FA_NotePreFunction` and `FA_NotePostFunction` notification for the book file, and for each document in the book. When trapping book-wide functions, you should check the value of `docId` to determine whether it indicates a document or the active book.

For example, if you search a book with two documents in it, the FrameMaker product posts the following function notifications:

```
FA_Note_PreFunction (start searching book)
FA_Note_PreFunction (start searching first document)
FA_Note_PostFunction (stop searching first document)
FA_Note_PreFunction (start searching second document)
FA_Note_PostFunction (stop searching second document)
FA_Note_PostFunction (stop searching book)
```

Miscellaneous book-related changes

The following functions have minor changes related to creating or processing books or book components:

F_ApiNewNamedObject()

You can create an untitled book if you pass `NULL` as the `name` argument. When you create a book in FrameMaker+SGML, the book is automatically structured.

F_ApiSimpleImportFormats()

This function can return the following additional values for `FA_errno`:

- `FE_Canceled` if the user canceled the operation
- `FE_FailedState` if the FrameMaker product failed to open one or more component documents during the import operation.

Changes to **F_ApiFind()**

`F_ApiFind()` now recognizes the `FS_FindWrap` scripting property; the default value is `True`. When this is `True`, the find operation continues to search past the starting location, returning information for matches it already found.

When `FS_FindWrap` is `False`, the find operation searches the document until it comes back to the starting location. From there it returns an empty `F_TextRangeT`, and `FA_errno` is set to `FE_NotFound`. This makes it easier to implement your own book-wide searches.

Changes to optimize PDF file size

When you saved a document as PDF in earlier versions, every paragraph and structure element was stored as a named destination. This ensured that links such as cross-references would work in the resulting PDF. However, the resulting PDF file size was very large.

Properties to identify named destinations

In version 6.0, as the user creates cross-references or hypertext links, the FrameMaker product marks the associated paragraph or structure element to be used as a named destination. Each paragraph and structure element now includes an `FP_MarkedForNamedDestination` property; `True` indicates that paragraph or element will be a named destination.

To indicate whether there are named destinations in a file, document and book objects include the `FP_DestinationsMarked` property. If this is `True`, the product will use the information stored in the `FP_MarkedForNamedDestination` properties. Your clients

should not set `FP_DestinationsMarked` to `True`. A value of `False` usually indicates that the document was created in an earlier version, but the document wasn't optimized via the `PDFSize` API client (see below).

API client to optimize legacy documents

Documents created in earlier versions do not automatically include information for named destinations, even after users save them with version 6.0. To include this information, users must process them with the `PDFSize` client via the `Optimize PDF Size` commands. Your clients can also invoke `PDFSize` via `F_ApiCallClient()`.

Specifying Document Info for PDF

FrameMaker documents now include the `FP_PdfDocInfo` property to store arbitrary PDF Document Info fields; this data is included in the resulting file when you save as PDF. This property is a list of strings that specify Document Info field names and values.

Specifying values for Structured PDF

`FO_Doc` objects now include the `FP_PDFStructure` property to indicate whether the FrameMaker product should generate Structured PDF when you save the document as PDF.

`FO_PgfFmt` objects include the `FP_PDFStructureLevel` property to specify what level in the hierarchy paragraphs of a specific format should be.

New functions

The following functions are new for version 6.0:

| Function | Purpose |
|--|---|
| <code>F_ApiUpdateBook()</code> | A scripted function to update books. |
| <code>F_ApiGetUpdateBookDefaultParams()</code> | Returns the default parameters to use with <code>F_ApiUpdateBook()</code> . |
| <code>F_ApiPrintUpdateBookStatus()</code> | Prints errors returned by <code>F_ApiUpdateBook()</code> . It is useful for debugging your clients. |

Modified objects

The following sections describe new or modified properties and property values for existing objects and object properties. Properties and values that have not changed from earlier releases are not listed here.

Book objects

The following are new or changed properties for a book object:

- FP_BookIsSelected
- FP_DocPDFColumnArticleThreads
- FP_DocPDFDefaultsChanged
- FP_DocPDFElementList
- FP_DocPDFElements
- FP_DocPDFNoArticleThreads
- FP_GeneratePDFInfo
- FP_PDFDestinationsMarked
- FP_PDFDocInfo
- FP_PFDAllNamedDestinations
- FP_BookIsViewOnly
- FP_TypeOfDisplayText
- FP_ViewOnlyDeadCodes
- FP_ViewOnlyWinBorders
- FP_ViewOnlyWinMenubar
- FP_ViewOnlyWinPopup

The following are obsolete properties for a book object (but retained for backward compatibility):

- FP_AcrobatBookmarkDisplayTags
- FP_DocAcrobatColumnArticleThreads
- FP_DocAcrobatDefaultsChanged
- FP_DocAcrobatElementList
- FP_DocAcrobatElements
- FP_DocAcrobatNoArticleThreads
- FP_GenerateAcrobatInfo

Book component objects

The following are new or changed properties for a book component object:

- FP_ChapNumComputeMethod
- FP_ChapterNumber
- FP_ChapterNumStyle
- FP_ChapterNumText
- FP_ComponentDisplayText
- FP_FirstPageNum
- FP_FnFirstNum
- FP_FnCustNumString

- FP_FnNumComputeMethod
- FP_FnNumStyle
- FP_PageNumComputeMethod
- FP_PgfNumComputeMethod
- FP_PageNumStyle
- FP_TblFnCustNumString
- FP_TblFnNumStyle
- FP_TblFnNumComputeMethod
- FP_VolNumComputeMethod
- FP_VolumeNumber
- FP_VolumeNumStyle
- FP_VolumeNumText

The following are obsolete properties for a book component object:

- FP_PageNumbering
- FP_PagePrefix
- FP_PageSuffix
- FP_PgfNumbering

As described above, some of the book component numbering properties have changed. In addition, the following constants for book component numbering properties have changed:

| Old value | New value |
|-----------------------|-----------------------|
| FV_BK_CONT_PAGE_NUM | FV_NUM_CONTINUE |
| FV_BK_RESET_PAGE_NUM | FV_NUM_RESTART |
| FV_BK_READ_FROM_FILE | FV_NUM_READ_FROM_FILE |
| FV_BK_CONT_PGF_NUM | FV_NUM_CONTINUE |
| FV_BK_RESTART_PGF_NUM | FV_NUM_RESTART |

Command objects

The following are new or changed values for the FP_EnabledWhen property of a command object:

- FV_ENABLE_NEEDS_DOCP_OR_BOOKP
- FV_ENABLE_BOOK_HAS_SELECTION
- FV_ENABLE_DOC_OR_BOOK_HAS_SELECTION

Document objects

The following are new or changed properties for a document object:

- FP_DocPDFColumnArticleThreads

- FP_DocPDFDefaultsChanged
- FP_DocPDFElementList
- FP_DocPDFElements
- FP_DocPDFNoArticleThreads
- FP_GeneratePDFInfo
- FP_PDFAllNamedDestinations
- FP_PDFBookmark
- FP_PDFBookmarkDisplayTags
- FP_PDFBookmarkDisplayTags
- FP_PDFDestinationsMarked
- FP_PDFDocInfo
- FP_PDFStructure
- FP_ChapNumComputeMethod
- FP_ChapterNumber
- FP_ChapterNumStyle
- FP_ChapterNumText
- FP_PageNumComputeMethod
- FP_PgfNumComputeMethod
- FP_VolNumComputeMethod
- FP_VolumeNumber
- FP_VolumeNumStyle
- FP_VolumeNumText
- FP_FnNumComputeMethod

The following are obsolete properties for a document object (some are retained for backward compatibility):

- FP_AcrobatBookmarkDisplayTags
- FP_DocAcrobatColumnArticleThreads
- FP_DocAcrobatDefaultsChanged
- FP_DocAcrobatElementList
- FP_DocAcrobatElements
- FP_DocAcrobatNoArticleThreads
- FP_GenerateAcrobatInfo
- FP_FnNumberingPerPage

Paragraph format objects

The following are new or changed properties for a paragraph format object:

- `FP_MarkedForNamedDestination`
- `FP_PDFLevel`
- `FP_PDFStructureLevel`

The following are obsolete properties for a paragraph format object (but retained for backward compatibility):

- `FP_AcrobatLevel`

Session objects

The following are new properties for a session object:

- `FP_GreekSize`
- `FP_Gravity`
- `FP_Snap`

Structural element objects

The following is a new property for a structural element object:

- `FP_MarkedForNamedDestination`

1

Modified objects

2 *Samples in the FDK*

Changes to Samples

The following changes have been made to the samples.

Windows®

Most of the samples now come with Microsoft® Visual C++ project files (.dsp and .dsw). These projects were created in Visual C++ 6.0. If you have a previous version of Visual C++, you may need to recreate the project files. See the *FDK Platform Guide: Windows* for more information.

Macintosh

The 68K platform is no longer supported.

CodeWarrior project files are included with most of the samples. These projects were created with CodeWarrior IDE 4.0. If you have a previous version of CodeWarrior, you may need to recreate the project file.

Compiling with MPW is supported with CodeWarrior MPW compiler version 4.0 or later.

See the *FDK Platform Guide: Macintosh* for more information.

New Samples for Version 6.0

With release 6.0, the following samples have been added to the FDK:

- SGMLPlus Development Environment
- includelists

SGMLPlus Development Environment

The SGMLPlus Development Environment is a C++ application that extends the functionality for writing SGML API clients. SGMLPlus is built on top of the SGML API; it is not a new API to the product.

The SGMLPlus sample demonstrates the following:

| Functionality | Description |
|---------------------------------------|---|
| Multiple event loops | <p>The SGMLPlus Development Environment allows you to define multiple SGML readers and writers.</p> <p>SGMLPlus allows you to have multiple reader-objects and writer-objects. Each reader object has it's own read-handler (<code>Handle_Sr_Event</code>) that takes the same parameters as the <code>Sr_EventHandler</code>. Each writer object has it's own writer handler (<code>Handle_Sw_Event</code>) that takes the same parameters as the <code>Sw_EventHandler</code>.</p> <p>Standard SGML API clients can define a single event loop for SGML read and SGML write. In a large application, the event loop can become cluttered with code, variable definitions, and flags. Multiple event loops allow you to isolate the code for a particular function of SGML read or write and to develop and test the functionality in isolation from other parts of your SGML read or write process.</p> |
| Custom rules in read/write rules file | <p>The SGMLPlus Development Environment shows one way to add custom rules to the read/write rules file defined for an application. For example, there is no read/write rule to wrap an element in another element. The sample shows how to create and use this rule.</p> |
| Multiple templates | <p>The SGMLPlus sample shows one way of switching templates based on element name.</p> <p>The <code>sgmlapps.fm</code> file allows you to define a single template for the application. Many applications need to switch the template based on the element. For example, the appendix element should be put in the template <code>appendix.fm</code>.</p> |

Functionality

Handling Index
markers

Rearrange elements

Description

This sample shows a C++ implementation of handling index markers.

The SGML read/write rules offer limited support for reading and writing index markers. Mapping between the DTD elements and the FrameMaker+SGML index marker representation often requires API customization. The docbook sample in the FrameMaker+SGML/sgml directory has another implementation of handling index markers.

In some cases, the order of elements in the DTD does not match how they should appear in FrameMaker+SGML. For example, the DTD often specifies that the Figure element contains the element Title followed by Graphic. However, Title should appear below Graphic in FrameMaker+SGML. The sample rearranges these two element on SGML read and SGML write.

Includelists

FrameMaker 6 removed the shuttle dialogs from the Print, Import Formats, and Import Element Definitions dialogs. These dialogs now apply changes to the current selection in the book window.

If the user wants the pre-FM 6 behavior, he can use the MIF statements `DefaultPrint` and `DefaultApply` to specify which files in a book will be affected by the command. Then, he can use the menu items defined by this client.

File>Print Files in Include List...

Use this menu item if you want to print the files in a book whose `DefaultPrint` MIF statement is Yes (or whose `FP_PrintInclude` property is set).

File>Import>Formats to Files in Include List...

Use this menu item if you want to import formats to the files in a book whose `DefaultApply` MIF statement is Yes (or whose `FP_ImportFmtInclude` property is set).

File>Import>Element Definitions to Files in Include List...

Use this menu item if you want to import formats to the files in a book whose `DefaultApply` MIF statement is Yes (or whose `FP_ImportFmtInclude` property is set).

3

Last Minute Information

This chapter contains last-minute information and tips on release 6.0 of the Frame® Developer's Kit (FDK) and updates to the FDK documentation for this release. For an overview of new FDK features in release 6.0, see [Chapter 1, "What's New in the FDK."](#)

Developing with the API on all platforms

Read the following before working with the Frame Application Programming Interface (API). The information and tips in this section apply to FDK clients on all supported platforms. Later sections cover platform-specific information.

FP_ElementSelection isn't properly registered when selecting multiple book components

If multiple book components are selected in a structured book and the book window is the frontmost window, the `FP_ElementSelection` property for the book will not provide a correct value.

However, if multiple book components are selected in a structured book, and the Structure View is the frontmost window, the `FP_ElementSelection` property for the book will provide a correct value.

A workaround is to define a keyboard shortcut and use the keyboard shortcut when the Structure View is the frontmost window.

Acrobat® property not defined for books

`FP_GenerateAcrobatInfo` isn't defined for books. This property corresponds to the "Generate Acrobat Data" checkbox in the print dialog.

Cannot import SGML fragments via `F_ApiImport()` without displaying dialog

It is not possible to import SGML fragments into FrameMaker+SGML without displaying the "Import SGML text by reference" dialog.

```
/* This is the property to allow for import of SGML fragments
   via the API. Otherwise, SGML fragments are NOT recognized,
   and F_ApiImport returns FE_FailedState */
i = F_ApiGetPropIndex(&props, FS_ImportAsType);
props.val[i].propVal.u.ival = FV_TYPE_SGML;
/* Dialog still appears when importing SGML fragments via
   F_ApiImport() */
i = F_ApiGetPropIndex(&props, FS_ManualUpdate);
```

```
props.val[i].propVal.u.ival = False;
```

The above scenario involves the programmatic import of SGML fragments into a structured FrameMaker+SGML document. Setting the property `FV_TYPE_SGML` allows the SGML fragment to be recognized, then the "Import SGML text by reference" dialog appears, asking for the user to choose 'manual' or 'automatic' update.

Setting the property `FS_ManualUpdate` to `False` has no effect (i.e., the dialog still appears). Setting other properties like `FS_HowToImport` to `FV_DoByRef` has no effect either.

F_ApiDeletePropByName()

If you call the `F_ApiDeletePropByName()` function to delete a facet of a graphic inset, the function does not delete the facet, although the function returns `FE_Success`.

FO_Arc object and objects representing Bezier curves

For `FO_Arc` objects and Bezier curves represented by `FO_Line`, `FO_Polygon`, and `FO_Polyline` objects, Frame products may show very poor performance if you specify large values to describe the bounding box of the object. Make sure that each of the following metric values is less than 150 inches (10,800 points):

- `FP_LocX`
- `FP_LocY`
- The sum of `FP_Width` and `FP_LocX` (`FP_Width+FP_LocX`)
- The sum of `FP_Height` and `FP_LocY` (`FP_Height+FP_LocY`)

Traversing a list of invalid elements

When traversing a list of invalid elements, it is possible that your client will enter an endless loop. This can happen when you use `FP_HighestLevelElement` for the `FO_Flow` object, and loop from that element using `FP_NextInvalidElement`.

To work around this problem, use `FP_HighestLevelElement` as before and traverse the tree of elements using `FP_FirstChildElement` and `FP_NextSiblingElement`. Then get `FP_ValidationFlags` for each element in the tree.

Developing with the API on Macintosh platforms

The information and tips in this section are specific to Macintosh FDK clients.

F_Sprintf()

Do not pass `RealT` or floating-point values to `F_Sprintf()` or other functions that use a variable number of arguments. This may result in problems with your FDK client.

Developing with the API on Windows platforms

The information and tips in this section are specific to Windows FDK clients.

FA_Note_BackToUser notification

Two `FA_Note_BackToUser` notifications are sent when the user clicks the following in the Tools palette:

- Smart Selection tool
- Object Selection tool
- Fill pop-up menu
- Pen pop-up menu
- Color pop-up menu
- Line Styles pop-up menu
- Line Ends pop-up menu
- Line Widths pop-up menu

The additional `FA_Note_BackToUser` notifications occur after the normal `FA_Note_PreFunction` and `FA_Note_PostFunction` notifications.

Labels with keyboard shortcuts

When you compile a custom dialog box that has labels with keyboard shortcuts, the compilation process adds ampersands to the label names to identify the keyboard shortcuts. For example, a label that appears as QK (with the letter *O* underlined) in the DRE file becomes &OK (with the ampersand preceding the letter *O*) after the compilation process.

If you use the label name in the code of your FDK client, be sure to take the ampersand into account.

Also note, to specify keyboard shortcuts for menu commands, you use the same syntax. For example, assume you have the following menu command: MyApiCommand (with the letter *C* underlined). The menu string must be specified as MyApi&Command.

Designing dialog boxes in Frame products

When editing DRE files in Frame products, keep the following in mind.

Editing DRE files in UNIX

When you edit a DRE file in a UNIX version of a Frame product, if you change the item order in the DRE file while testing it, the appearance of the dialog box may not match its appearance in the DRE file. To solve this problem, exit the Frame product and start a new Frame session.

Highlighting text in multiline text boxes

When running Windows clients with custom dialog boxes, you may notice that text in the multiline text box remains highlighted, even if another item is active.

Relating image pop-up menus and radio buttons

For Windows FDK clients, if you relate an image pop-up menu to a radio button, selecting the image pop-up menu does not cause the radio button to be selected.

Resetting a platform view in a DRE file

After you change to a platform view, you can either modify the platform view, which causes the Frame product to cache the modified view, or you can reset the view. You cannot do both. A change in the layout of a view causes the Frame product to cache the view and to ignore any commands to reset the view.

For example, if you edit the UNIX view of a DRE file and press `Esc d r` to reset the view, the view is not reset. Your changes to the layout cause the Frame product to cache the modified view and to ignore the command to reset the view.

Before you attempt to reset a view, make sure that you have made no layout changes. You can do this by using the keyboard shortcut for changing to the current view before resetting the view. Because you make no changes between the time you change to the current view and the time you reset the view, the Frame product can reset the view.

For example, if you are editing the UNIX view of a DRE file, follow these steps to reset the UNIX view:

1. Press `Esc d x` to change to the current view.
2. Press `Esc d r` to reset the view.

Testing a dialog box in Windows NT

In some cases, items in a DRE file may appear larger in the actual dialog box. This may cause items to overlap. For example, if you place a radio button to the left of a text box, the label of the radio button may overlap the text box in the actual dialog box.

Test your dialog boxes within the Frame product to make sure that items do not overlap.

Also, if you are testing a dialog box and notice that the displayed dialog box is truncated, set the `MonitorSize` entry in your `maker.ini` file to `0in`. This should correct problems in the display of dialog boxes.

Updates to the FDK documentation

This section describes updates to the FDK documentation in this release.

FDK Programmer's Guide and Reference

This section describes updates to the online manuals *FDK Programmer's Guide* and *FDK Programmer's Reference*.

Aside from documenting the changes to the FDK, the *FDK Programmer's Reference* has been reviewed, and a number of errors have been corrected. The *FDK Programmer's Guide* now includes a chapter discussing how to process books.

SGML API Programmer's Guide

This section describes updates to the online manual *SGML API Programmer's Guide*.

The *SGML API Programmer's Guide* has been reviewed, and many errors have been corrected. Also, the discussions about how the SGML API works, how to write programs with it, and how the entity handler works have been improved. Finally, many of the code examples in the manual have been improved.

Book component events and document events

The SGML API means something different by the term *book component* than does the rest of the FDK.

In standard FDK clients, if you work with a book's components, you are only working with the data structure in the book file itself. You are not working with the file or contents of the book component. In the SGML API, however, a book component refers to the actual book component file.

DocBook clients

The third chapter of the *SGML API Programmer's Guide* presents the code for the DocBook Starter Kit's SGML clients, and provides details about the code. However, these are intended to be examples only. They do not match the latest version of the DocBook Application Pack.

