



# FDK 7.0 Release Notes

Adobe



© 2002 Adobe Systems Incorporated. All rights reserved.

#### Adobe FrameMaker 7.0 FDK Release Notes

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.



# Contents

<b>Chapter 1</b>	<b>What's New in the FDK</b>	<b>1</b>
What's New for Version 7.0		1
Changes to the FDK documentation		1
Only one FrameMaker product		2
Import and export of XML		2
Enhanced text encoding which includes UNICODE and ICU libraries		3
XMP metadata for workflow management		4
Support for WebDAV		4
Changed FDK functions		4
New and changed FDK properties		5
What's New for Version 6.0		8
Book window changes		8
View-only books		8
Book error log		9
Numbering for documents and book components		9
Scripted book updates		9
Book-wide command processing		10
Miscellaneous book-related changes		10
Changes to F_ApiFind()		11
Changes to optimize PDF file size		11
Specifying Document Info for PDF		11
Specifying values for Structured PDF		11
New functions		12
Modified objects		12
<b>Chapter 2</b>	<b>Last Minute Information</b>	<b>17</b>
Developing with the API on all platforms		17
Developing with the API on Windows platforms		18
Designing dialog boxes in Frame products		19
Editing DRE files in UNIX		19
Highlighting text in multiline text boxes		19
Relating image pop-up menus and radio buttons		19
Resetting a platform view in a DRE file		20
Testing a dialog box in Windows		20



# 1

## What's New in the FDK

This chapter lists changes to the FDK that are the result of changes to the FrameMaker product. The chapter lists a history of changes, beginning with release 6.0. The list of changes is in descending order, starting from the most recent release.

---

### What's New for Version 7.0

With release 7.0, Adobe FrameMaker® introduce the following, which impact the FDK:

- Changes to the FDK documentation
- Only one FrameMaker product—a user can run FrameMaker in structured or unstructured mode
- Import and export of XML
- Enhanced text encoding which includes UNICODE and ICU libraries
- XMP metadata for workflow management
- Support for WebDAV
- More Header/Footer variables
- New or Changed FDK properties

### Changes to the FDK documentation

Aside from updates for this new release, there have been slight changes to the documentation. Most of the FDK documentation has a new layout, which is more consistent with other Adobe Systems SDK documentation.

An appendix was added to the FDK Programmer's Reference that describes how to use `F_ApiCallClient()` with the clients that ship with the FrameMaker product. This information used to be in the discussion for the function, `F_ApiCallClient()`.

Because there is only one FrameMaker product, function descriptions no longer include a list of FrameMaker products that work with the function. Instead, the functions that require structured FrameMaker are identified at the beginning of the function description.

The third chapter of the *Structure Import/Export API Programmer's Guide* presents the code for the XHTML Starter Kit's import/export client, and provides details about the code. In previous versions this chapter used a minimal example of the DocBook starter kit client.

The FDK documentation now includes *FDK Sample Clients*, a list of sample clients in a the file, `samplelist.pdf`.

## Only one FrameMaker product

With release 7.0, Adobe Systems merged FrameMaker and FrameMaker+SGML into the same product. The user specifies which product interface to run with FrameMaker via user preferences. Also, there is no 7.0 release of FrameViewer or FrameReader.

### Registering FDK clients

In earlier versions you could specify which FrameMaker products could load your FDK client. In FrameMaker 7.0 you can still specify which product interface is required for a session in order for FrameMaker to load your clients. For Unix and Windows, you specify this via client registration. For the Macintosh, you can query the `FP_ProductIsStructured` session property, and cancel client initialization if necessary. For more information, see the sections on registering clients in the FDK platform guides.

### Determining the product interface for a session

To determine the current product interface, get the `FP_Platform` session property. The value for this property can be one of:

- `FrameMaker`
- `FrameMaker+SGML`
- `FrameViewer`
- `DemoMaker`
- `DemoMaker+SGML`

These are the same values as for the 6.0 release of FrameMaker. `FrameViewer` is saved for backward compatibility, and `FrameMaker+SGML` or `DemoMaker+SGML` indicate that Framemaker is running in structured mode.

## Import and export of XML

Release 7.0 of FrameMaker adds import and export of XML to the product when running in structured mode. Many things have changed as a result.

### Name changes in the documentation

- The SGML API has been renamed the Structure Import/Export API
- SGML applications have been renamed structure applications
- The *SGML Application Developer's Guide* has been renamed the *Structure Application Developer's Guide*
- The `sgml` application file is now called the structure application file—its filename has changed from `sgmlapps.fm` to `structapps.fm`

### Changes to naming in the FDK and Structure Import/Export API

In release 6.0 many functions, properties, and values included “SGML” in their names. Because the API now processes XML as well as SGML, these naming conventions aren't

strictly correct. Naming within the Structure Import/Export API has changed occurrences of “SGML” to “Structured”.

For backward compatibility, all the old names are retained. However, these names are deprecated in preference to the new names. The documentation for the Structure Import/Export API shows the deprecated names for functions, but not for properties and values. For legacy code, you should not have to rewrite the code using new names. However, for new code you are encouraged to use the new naming.

For a list of new and changed FDK properties, see [“New and changed FDK properties” on page 5](#).

### Changes to link settings

To use the Structure Import/Export API in a client, you must link in the Xerces library for XML parsing, and the ICU library for processing text encodings. These libraries ship with the FrameMaker product—you must link to these same libraries. For information on the proper link settings, see the FDK platform guides.

## Enhanced text encoding which includes UNICODE and ICU libraries

FrameMaker now supports UNICODE text for import and export of XML. Upper-range double-byte characters can occur in XML element content and in XML tokens such as element GIs and attribute names.

FrameMaker ships with support for the following encodings, listed by their IANA names:

Big5	KSC_5601
EUC-CN	macintosh
EUC-JP	Shift_JIS
EUC-KR	US-ASCII
EUC-TW	UTF-16
GB2312	UTF-8
ISO-8859-1	windows-1252

These encodings are created in the ICU (International Components for Unicode) format, and stored as .cnv files. The supplied encodings are stored in the following locations:

- Windows: \$installdir\fminit\icu\_data
- Unix: \$installdir/fminit/icu\_data
- Macintosh: \$installdir:Modules:Icu\_Data

Users can add other ICU encodings to the FrameMaker installation—to do this they create ICU compliant mappings and save them as .cnv files, then store them in the icu\_data directory. You can refer to Codings installed by a user the same as you refer to the encodings that are shipped with FrameMaker..

For more information about how FrameMaker handles UNICODE and ICU libraries, see Chapters 5 and 7 in the *Structure Application Developer's Guide*. For more information about ICU libraries, see the ICU web site at <http://www-124.ibm.com/icu/>.

## XMP metadata for workflow management

The eXtensible Metadata Platform (XMP) provides a common XML framework that standardizes the creation, processing, and interchange of document metadata across publishing workflows. For more information about XMP and access to the XMP SDK, see the XMP web pages at <http://www.adobe.com/products/xmp/main.html>.

For a FrameMaker document or book, XMP data is stored in the `FP_FileInfoPacket` property—you can use the FDK `get` and `set` the value of this `StringT`. For more information, see the discussions for XMP Metadata in the *FDK Programmer's Reference* for `FO_Document` and `FO_Book` objects.

## Support for WebDAV

The FrameMaker product includes support for WebDAV (World Wide Web Authoring and Versioning), which includes connecting to WebDAV servers, checking files in and out, and updating files. This support is provided by an FDK client named WebDAV. For information about calling this client from your own clients, see Appendix A of the *FDK Programmer's Reference*.

The FDK includes various properties that are used by the WebDAV client, but your clients should not use these properties to manage WebDAV interactions. Instead, you should use the WebDAV client.

## Changed FDK functions

### Opening, importing, and saving files

The following functions use or get property lists to script function behavior when performing file I/O with FrameMaker documents:

- `F_ApiGetImportDefaultParams()`
- `F_ApiGetOpenDefaultParams()`
- `F_ApiGetSaveDefaultParams()`
- `F_ApiImport()`
- `F_ApiOpen()`
- `F_ApiSave()`

These functions use the following new properties and values to manage file I/O with XML:

- `FS_DisallowXml`
- `FS_FileIsXmlDoc`



- FS\_FileType
  - FV\_SaveFmtBinary60
  - FV\_SaveFmtXml
- FS\_ImportAsType and FS\_OpenAsType
  - FV\_TYPE\_XML
- FS\_OpenNativeError and FA\_errno
  - FV\_CancelFileIsXml
  - FV\_CancelFileXml
  - FV\_ImportedXmlDoc

### Notification constants

The following constants have been added for use with `F_ApiNotification()` and `F_ApiNotify()`.

- FA\_Note\_PreOpenXML
- FA\_Note\_PostOpenXML

## New and changed FDK properties

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

### Session properties

- FP\_ProductIsStructured
- FP\_FM\_StructureDir
- FP\_FM\_XmlDir

### Book and document properties: PDF

- FP\_FileExtensionOverride
- FP\_FileInfoPacket
- FP\_PDFJobOption
- FP\_PDFOpenPage
- FP\_PDFZoomType
  - FV\_PDFZoomNone
  - FV\_PDFZoomDefault
  - FV\_PDFZoomPage
  - FV\_PDFZoomWidth
  - FV\_PDFZoomHeight
  - FV\_PDFZoomMaxValue

- FP\_PDFRegistrationMarks
  - FV\_PDFRegistrationMarksNone
  - FV\_PDFRegistrationMarksWestern
  - FV\_PDFRegistrationMarksTombo
  - FV\_PDFRegistrationMarksMax
- FP\_PDFZoomFactor
- FP\_PDFSeparateFiles
- FP\_PDFPageWidth
- FP\_PDFPageHeight
- FP\_PDFPrintPageRange
- FP\_PDFStartPage
- FP\_PDFEndPage
- FP\_PDFConvertCMYKtoRGB
- FP\_PDFBookmarksOpenLevel
  - FV\_PDFBookmarksOpenDefaultLevel
  - FV\_PDFBookmarksOpenAllLevels
  - FV\_PDFBookmarksOpenNoneLevel
- FP\_PDFDistillerAbsent

#### **Book and document properties: XML**

- FP\_XmlVersion
- FP\_XmlEncoding
- FP\_XmlStandAlone
  - FV\_XML\_STANDALONE\_YES
  - FV\_XML\_STANDALONE\_NO
  - FV\_XML\_STANDALONE\_NA
- FP\_XmlStyleSheet
- FP\_XmlStyleSheetList
- FP\_XmlUseBOM
  - FV\_XML\_USEBOM\_YES
  - FV\_XML\_USEBOM\_NO
  - FV\_XML\_USEBOM\_NA
- FP\_XmlWellFormed
  - FV\_XML\_WELLFORMED\_YES
  - FV\_XML\_WELLFORMED\_NO
  - FV\_XML\_WELLFORMED\_NA

- FP\_XmlFileEncoding
- FP\_XmlDocType
- FP\_XmlPublicId
- FP\_XmlSystemId

#### **Book and document properties: Header/Footer variables**

- FP\_SystemVar
  - FV\_VAR\_HEADER\_FOOTER\_5
  - FV\_VAR\_HEADER\_FOOTER\_6
  - FV\_VAR\_HEADER\_FOOTER\_7
  - FV\_VAR\_HEADER\_FOOTER\_8
  - FV\_VAR\_HEADER\_FOOTER\_9
  - FV\_VAR\_HEADER\_FOOTER\_10
  - FV\_VAR\_HEADER\_FOOTER\_11
  - FV\_VAR\_HEADER\_FOOTER\_12

#### **Table properties**

- FP\_TblInLockedTi

## What's New for Version 6.0

With release 6.0, Adobe 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, FrameMaker 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 structured FrameMaker, 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 FrameMaker 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 crossreferences 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, FrameMaker 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 FrameMaker 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



# 2

## Last Minute Information

This chapter contains last-minute information and tips on release 7.0 of the FDK. For an overview of new FDK features in release 7.0, see [Chapter 1, “What’s New in the FDK”](#).

---

### Developing with the API on all platforms

Read the following before working with the FDK or the Structure Import/Export API. The information and tips in this section apply to FDK clients on all supported platforms. Later sections cover platform-specific information.

#### **When importing XML or SGML, you cannot save a document until the document is completely imported**

Developers have reported a crash when saving a document during import of XML or SGML. This happens when a client uses the `SR_EVT_END_DOC` parsing event to trigger a save operation. Although the parser is finished with the document at this moment, the FrameMaker document is not yet complete—trying to save a document in this state results in a crash.

The recommended way to save a document after importing SGML or XML is to use `F_ApiNotification()` to request a notification for a `FA_Note_PostOpenSgml` or `FA_Note_PostOpenXml` event through the FDK. Then use the notification handler to save the document.

This issue is also noted in the *Structure Application Developer’s Guide*.

#### **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 loop from the root element of a flow (`FP_HighestLevelElement`), 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.

#### **Unstraddling table cells**

`F_ApiUnstraddleCells()` results in display problems in the affected table after you use it to unstraddle the cells. After using this function, the newly unstraddled cells have no end of flow markers, and the user cannot get an insertion point in the cells. If an entire row was straddled, subsequent rows in the table may “disappear” after unstraddling that row.

If you unstraddled less than an entire row, you can coerce FrameMaker to redraw the table, which should fix the problem. If the user types in the paragraph containing the table anchor, or if your FDK client changes the height of the affected table row, FrameMaker will redraw the

table and display the cells correctly. For example, the following calls add a paragraph to the unstraddled cell, then deletes it:

```
. . .
pgfId = F_ApiGetId(docId, cellId, FP_LastPgf);
pgfId = F_ApiNewSeriesObject(docId, FO_Pgf, pgfId);
if(pgfId) F_ApiDelete(docId, pgfId);
. . .
```

However, if you unstraddle a region that encompasses an entire row, this type of coercion doesn't work. To unstraddle an entire row or rows, you should:

- Determine the number of rows that are straddled (get the `FP_CellNumRowsStraddled` property for the cell)
- Insert that number of new rows in the table
- Put the contents of the straddled cell into the first cell of the first new row
- Delete the straddled cell

---

## 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

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

### **Character sets and filenames**

FrameMaker uses its own character set for internal representation of strings, and this set is not identical to the Windows OS character set. This can cause a conflict when using FDE functions to operate on filenames that include upper-range characters. For example, the Windows character set uses `0xfc` for umlaut 'ÿ', while the FrameMaker character set uses `0x9f` for the same character. Assume a filename that includes this character. When calling

`F_DeleteFile()`, you must use the FrameMaker representation of this character. — when calling `F_FilePathProperty()` you must use the Windows representation.

### 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 `OK` (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: `MyApiCCommand` (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 FrameMaker, 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 FrameMaker and start a new FrameMaker session.

If your dialog box uses scroll lists, you might want to include the single-quote character in your scroll list strings. For example, you might want to use a single quote to represent length in feet. By default, FrameMaker converts the single-quote character to an escaped double-quote. This is because single quotes are not valid in filenames. To override this default, add the following statement to the dialog box resource file along with the other statements for the given scroll list:

```
Maker.dialog.bb.itemX.isEnc: noConv
```

where `itemX` specifies the item number — for example, `Maker.dialog.bb.item7.isEnc` specifies item 7. If you place this statement at the head of the resource statements for the dialog box item, that scroll list will display single-quotes correctly.

### 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

In some cases, items in a DRE file may appear larger in the actual dialog box than they do when editing the file in FrameMaker. 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 FrameMaker 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 for NT, Windows 2000 and Windows XP.