

Guide to the fmPython

Release 0.2.1

Stefan

Petrucha

December 3, 2003

Copyright © 2002 Stefan Petrucha

Permission to use, copy and distribute the `fmPython` and its accompanying documentation for any purpose is hereby granted, provided that the complete package only (program + documentation + all accompanying files) and without fee will be redistributed. The author makes no representation about the suitability of `fmPython` for any particular purpose. It is provided “as is” without expressed or implied warranty.

This software may contain bugs and you agree to use it on your own risk. The author takes no responsibility for any accidental damages it could cause.

Please direct all questions and bug reports to `fmpython@isnet.sk` or to Stefan Petrucha, Hlavna 18, 931 01 Samorin, Slovak Republic.

Adobe, Frame, FrameMaker, FrameViewer, FrameMaker+SGML and FDK are trademarks of Adobe Systems Incorporated.

Table of Contents

Table of Contents	3
Preface	8
Release notes	10
Release 0.2.1	10
Changes to fmPython.	10
Changes to this manual	10
Release 0.2.0	11
Changes to fmPython.	11
Changes to this documentation	11
Installation.	12
What do you need	12
fmPython	12
FrameMaker.	12
Python	13
Installation	13
MS Windows (9x/NT/2000/XP)	13
Mac OS 8, 9	13
Using fmPython.	15
Invoking fmPython	15
User interface.	16
Define script command	16

Execute script command	16
Diagnostic	16
Editing scripts	17
Indentation	17
Text encoding	17
Formating	17
What to do with fmPython	18
Standard Python	18
Accessing FrameMaker	18
Module maker	18
Attributes of objects	20
Attributes common to more classes.	21
Book and document	21
Book, document and session	25
Book, document, session and dialog box.	26
Book component and document	27
Character related objects	29
Graphic objects	30
Unanchored graphic objects	33
Menu related objects	33
Paragraph related objects	34
Referenced text objects	37
Table and table format	39
Specific attributes	42
Anchored frame	42
Graphic arc	43
Body page	43
Book	44
Book component	45
Table cell	47
Character format	50
Color	53
Combined font definition	54
Command	55
Condition format	57
Dialog resource	59
Dialog box	59
Dialog button	60
Check box	60

Edit box	61
Dialog image	61
Dialog label	62
Pop-up menu	62
Radio button.	63
Scroll bar	63
Scroll box.	64
Three-state checkbox	65
Document.	65
Structural element	82
Definition of structural element.	86
Ellipse	87
Text flow	88
Change of format.	89
Formatting rule	91
Formatting clause.	91
Footnote	93
Group of objects	94
Hidden page.	94
Imported graphic	95
Graphic line	95
Marker	96
Type of marker	97
Master page	97
Mathematic expression	98
Menu	99
Paragraph of text	99
Paragraph format	101
Polygon	103
Polyline	103
Rectangle	104
Reference page.	104
Rectangle with rounded corners	105
Table row.	105
Rubi	107
Format of ruling.	107
Frame product session	108
Column of text.	113
Table	114
Format of table.	116

Text frame	116
Graphic text object	118
Text imported by an extension module	118
Text imported from a flow	119
Text imported by reference	119
Table imported by reference	120
Unanchored frame	121
Variable	121
Variable format	122
Cross-reference	123
Cross-reference format	124
Text	124
Procedural objects	127
Functions	127
Functions of the module maker	127
TxLocation()	127
TxRange()	128
Methods	128
Methods of wrapper objects	128
addText()	128
applyLayout()	129
close()	129
copy()	129
cut()	130
delete()	130
display()	130
execF()	131
foundIn()	131
getNamed()	132
Import()	132
newAnchor()	132
newGraphic()	133
Open()	133
paste()	134
Print()	134
Save()	134
select()	135
sOpen()	135
sUpdate()	135
text()	136

typeCode()	136
Update()	137
Symbolic values.....	138
Fill patterns	138
Grouped symbols.....	139

Preface

Programming is about managing complexity: the complexity of the problem you want to solve, laid upon the complexity of the machine in which it is solved. Because of this complexity, most of our programming projects fail. And yet, of all the programming languages of which I am aware, none of them have gone all-out and decided that their main design goal would be to conquer the complexity of developing and maintaining programs. I take this back on the 2nd edition: I believe that the Python language comes closest to doing exactly that. See www.Python.org.

[Bruce Eckel: Thinking in Java, 2nd edition]

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

[Guido van Rossum, Fred L. Drake, Jr.: Python Tutorial]

The fmPython is a way to use the Python interpreter from and within Adobe FrameMaker product family.

Why and for what purpose?

Please let me shortly introduce myself first. I've been heavily developing custom plug-ins for FrameMaker since 1996 with FDK. I did feel considerable lack of a general purpose tool for making pilot projects, application prototypes and proofs of concepts for FrameMaker effectively. Last but not least everybody needs to tweak or automatize some small thing here and there, in which case the FDK is just too big and mostly far away from what is really needed. For these cases scripting languages are appropriate.

There is FrameScript™/ElmScript™, but it is better suited for power users of FrameMaker and hobby programmers. It features proper problem-oriented scripting language in a nice package making the use easy. On the other hand it lacks the power of a general purpose programming tool and the possibility to interact with other programs or computers is very restricted.

Chapter 1

Release notes

1.1 Release 0.2.1

This is mainly a bugfix, but some new functions were added too.

1.1.1 Changes to **fmPython**

- Fixed bug that prevented to assign value to attributes representing points.
- Fixed bug that caused false syntax errors at script end in some contexts.
- Methods added: `display()`, `delete()` (for `TxRange` objects), `close()`.

1.1.2 Changes to this manual

- Fixed erroneous descriptions of attributes: `activeDoc` attribute (on page 108).

- A warning added to the chapter about installation, section for users of Macintosh computers.
- This chapter was a little bit restructured.
- There is a true table of contents now.

1.2 Release 0.2.0

1.2.1 Changes to fmPython

- Moved to the Python versions 2.2.x (Windows) and 2.2.1 (Macintosh).
- Text formatting ability added. Now it is possible to change properties of text through attributes of the TxRange object.
- New methods added, mainly for input/output operations.

1.2.2 Changes to this documentation

- Short section on what is in the module `maker` added.
- Almost all accessible functions are described.
- Almost all accessible symbolic values are described.
- Many new hyperlinks were added.

Chapter 2

Installation

2.1 What do you need

2.1.1 fmPython

You can find fmPython and its accompanying documentation online on the Web site at <http://www.isnet.sk/petrucha/>. There are distribution packages for supported platforms, FrameMaker and Python versions, available for download.

2.1.2 FrameMaker

The fmPython is not a standalone program. It is a plug-in for FrameMaker and as such it needs this commercial program to be operational. It does not matter whether you have the FrameMaker, FrameViewer or FrameMaker+SGML. More important is your Frame product has the desired version number. It is not a must, but there are some changes in internal structure of FrameMaker between versions. If the versions of both don't match, there will be some

features not fully functional, but the combination should generally cooperate.

2.1.3 Python

Nor is the `fmPython` a fully featured interpreter. It only interfaces an existing Python interpreter from Python Software Foundation. Now, the version does matter. The interpreter must be compiled as a shared library to be usable by `fmPython` and the version number must match. Otherwise the shared library will be not to find and the scripting impossible.

However, there is no Python interpreter at the `fmPython` web site. It must be downloaded separately from the Python web site at <http://www.python.org>. Whenever not stated differently, `fmPython` has been tested against the official releases available there.

2.2 Installation

This manual does not describe how to install all three necessary parts. There are other documents covering installation of `FrameMaker` and `Python`. It is supposed you have both correctly installed and operational already.

There is no installation program for `fmPython`. It is distributed as an archive created with an utility usual for that platform. You have to unpack it somewhere and move (or copy) the plug-in itself into the directory where `FrameMaker` looks for plug-ins. Then start or restart your `Frame` product. If all went good you should find the following message in the `FrameMaker` Console window: "`fmPython - Python scripting for FrameMaker`".

2.2.1 MS Windows (9x/NT/2000/XP)

On this platform `FrameMaker` searches for plug-ins in the `fminit\Plugins` subdirectory of its installation directory and deeper. The plug-in file itself is named `fmPython*.dll`, where `*` is wildcard for possible version number.

There is an older variant how to install a plug-in on this platform. I will not discuss it here even when it has some additional potential. Savvy users know what am I talking about.

2.2.2 Mac OS 8, 9

On this platform `FrameMaker` searches for plug-ins in the `Modules` subfolder of the `FrameMaker` installation folder. The plug-in file itself is named `fmPython *`, where `*` is wildcard for version number of the Python interpreter.

As the standard Macintosh edition of the Python interpreter as distributed is not compatible with FrameMaker, we prepared a patched version of the interpreter core. The fmPython does not run without this patched core library. The new core is backwards compatible with the original one - there should be no problems.

The old core has name PythonCore and is located in the installation folder of Python. Backup it (for example by renaming it to PythonCoreOriginal) and install the new PythonCore to the same folder. In the Extensions subfolder of the MacOS installation folder there is an alias to the old core. Its name is PythonCore *, where * is wildcard for the version of Python interpreter which must match with the version in name of the fmPython plug-in. After installing the new core redirect this alias to it (Command-I, ...).

For those that just do not trust to obscure patches there is file CoreDifferences.txt included in the fmPython distribution package containing all differences made to the standard PythonCore. Anybody can patch the source and compile the core herself. Maybe some day the patch reaches the standard core or somebody makes FrameMaker for OS X, in which case the patch will become unnecessary.

Having fmPython installed you may want to accomodate memory requirements of FrameMaker. fmPython runs in its memory space and a hungry script can fill it in. When this happens FrameMaker may end prematurely.

Warning

Python for the Mac OS 8-9 may contain two distinct interpreter flavours: classic and carbonized. fmPython is compatible with the classic version only. Remember this when your FrameMaker enriched with fmPython crashes right at start. Cure is quite simple: reconfigure Python to the classic version.

Chapter 3

Using fmPython

3.1 Invoking fmPython

Once properly installed, the fmPython starts with the FrameMaker session. The first signal it is ready is a message issued to the FrameMaker Console window, the second can be the menu fmPython on the main menu bar in some contexts.

There is even nothing special to shut it down - it is done so whenever the FrameMaker session ends. However, if you are skilled enough to run script with an endless loop, or you are not patient enough to wait until a long running script ends, there is currently no possibility to break it on MS Windows. You must use utils of the underlying operating system to shoot FrameMaker down. On the Macintosh, scripts can usually be interrupted with the well known key combination Command-. (Command-period).

3.2 User interface

The only user interface extension added to the session is new menu `fmPython` on the main menu bar for documents and books. Its unique identifier is `fmPython.menu`. This identifier is only then important for you, when you want to modify the visible menu label or hang it somewhere else in the menu tree.

There are two new commands in the menu. They are described below, but principle of interaction is: you define an active script and run it in the contexts and as many times as you need. At any given time there can be at most one active script.

3.2.1 Define script command

This command defines the active script. Its unique identifier is `fmPython.takeScript` and label `Selection as script`. Its keyboard shortcut is empty. You can change the label or keyboard shortcut with a command customization file to suit your aesthetic or other needs. If you do not know how to do it please read the [Customizing FrameMaker Products] online manual for your platform. Another possibility is to use a script written in Python ;-)

The command should be enabled only when the cursor is in text or there is some text selected. (I noticed it is erroneously enabled when a book is active, too.) If there is a selection, the command takes the text content of it for active script. If there is cursor only, contents of the whole text flow will be taken. The script is NOT checked for validity.

3.2.2 Execute script command

This command executes the active script. Its unique identifier is `fmPython.execScript` and label `Execute script`. Its keyboard shortcut is empty. The command should only be enabled when an active script was defined already.

All scripts are executed in the global namespace.

3.2.3 Diagnostic

The Python interpreter is a terminal oriented program. Therefore it obviously uses standard input, output and error output streams. Errors caught by the interpreter or runtime system are printed to the standard error output.

On the other hand FrameMaker is typical windowing program with no standard streams. Fortunately, it features the so called FrameMaker Console window, where some of its messages appear. It is just there where you will find diagnostic output issued by `fmPython` and `Python`.

If your script simply does nothing, check the `FrameMaker Console` window for possible error messages.

The `FrameMaker Console` has a feature of writing the content simultaneously into a file. This file can be very useful when you are forced to shoot down `FrameMaker` whenever it is running a script. The console window is discarded together with the `FrameMaker`'s one, but there is a good chance you will see the diagnostic output written to this file. For example on MS Windows the name of the console's output file is `console.txt` and it is located in the installation directory of `FrameMaker`.

3.3 Editing scripts

When editing scripts in `FrameMaker`, several things should be taken into account.

3.3.1 Indentation

In the `Python` language the indentation of program lines is used to group statements into blocks. Therefore it is wise to use monospaced fonts for your scripts, and left aligned paragraph formats.

When editing scripts, spaces and tabulators are generally used for indentation. But in `FrameMaker` there is no invariant relation between width of space character and size of a tabulator, and tabulator size is not fixed. You should take this into account and better not to mix spaces and tabulators in your scripts.

3.3.2 Text encoding

`FrameMaker` does not use standardized character encoding. At this stage of `fmPython` development there is no codec built into it to make the conversions easier. Don't use non-ASCII characters for program code. Special attention should be taken for quotes - switch off smart quotes and use only ASCII single and double quotes in documents containing scripts. Do not use special typographic spaces (thin, number, m, n, ...). Even tabulators are currently unsupported (use only spaces for indentation).

Usage of nonstandard characters in literals should be safe.

3.3.3 Formatting

Automatic line breaks are ignored and hyphenated probably too - such lines are considered unbroken. Backslashes at paragraph and forced line ends should serve as line binding characters.

Character formats and other format changes are ignored. This can be used for example for syntax coloring. However, you must do it yourself as this feature is not implemented. If you use non-text font, your program will be not very human-readable ;-)

3.4 What to do with **fmPython**

3.4.1 Standard Python

With **fmPython** you have the complete Python language and standard library at your fingertips in **FrameMaker**. Longing to turn it into Web server? Yes, you can!

But there are sure better things to do with it.

If you are not familiar with the Python language, please read the [Python Tutorial] first (which is part of the Python distribution). It is advisable to make your very first experiences directly with the Python program (the bare interpreter or IDE), as there is no such dialog mode possible (or better not so comfortable) with **fmPython**.

3.4.2 Accessing **FrameMaker**

It would be just useless to run ordinary scripts from within **FrameMaker**. Crucial is to be able to access and manipulate things available in it: documents and books with their contents. **fmPython** provides some necessary parts to do so. All **Frame**-related extensions to standard Python are held in the module **maker**. Through **fmPython** this module is built into the Python interpreter. It is possible that this module is already imported just after start of the session; you may import it otherwise.

3.4.3 Module **maker**

Directory of this module is not very big: there are objects **ConsoleLog**, **symbol** and **session** and functions **TxLocation()** and **TxRange()**. Later, after performing an input/output operation, **ioStatus** variable can be added.

The **ConsoleLog** object redirects the standard and error output of Python to the **FrameMaker Console** window. It is assigned to **sys.stdout** and **sys.stderr** when **fmPython** starts.

There is plenty of constants used across the API of **FrameMaker**. Their values say nothing about the meaning so it's better to give them somewhat more saying names. The **symbol** object serves

as a bridge to these symbolic values. So if for example `loc` is a `TxLocation` object, you can import an image in the MIF format to the location as

```
inset = loc.Import("patterns.mif", \
    {maker.symbol.HowToImport:maker.symbol.ImportByCopy})
```

instead of the less readable

```
inset = loc.Import("patterns.mif", {4:6})
```

Enemies of extensive writing can assign the `maker.symbol` object to another variable with shorter name and use this shortcut:

```
sym = maker.symbol
...
inset = loc.Import("patterns.mif", \
    {sym.HowToImport:sym.ImportByCopy})
```

The `session` object is a “window” into interior of `FrameMaker`. Through its attributes you have access to properties of the running `FrameMaker`, some of which reference to other internal object with their properties etc.

After performing an input/output operation, such as open/import/save, additional informations about it in the form of list of strings is assigned to `ioStatus`. The value remains there until next such operation.

Chapter 4

Attributes of objects

Properties of object parts of `FrameMaker`'s internal structure are directly accessible through attributes of wrapper (Python) objects. The wrapper objects have no other attributes. This chapter contains reference of all object attributes.

When getting attribute values exceptions may occur that are not considered as errors. They are generated when a value is missing.

A few value types are still unimplemented - mainly those related to structural extensions (SGML, XML) of the former bare `FrameMaker`. You can recognise them by their type of value `\N` (not a real type).

`TxLocation` and `TxRange` objects have the same set of attributes but both behave a little bit differently. All attributes of `TxLocation` objects are read-only and describe properties of character to the right of the location. All attributes of `TxRange` objects are write-only and the assigned value applies to all characters within the range. However, several attributes are from the nature read-only and these do not work for `TxRange` objects.

4.1 Attributes common to more classes.

4.1.1 Book and document

Table 1: Book, Document have the following attributes.

Attribute	Type of value	Meaning
<code>allTargets</code>	integer	True if bookmarks are to be generated for all possible targets (all paragraphs and eventually structural elements).
<code>application</code>	string	Name of the SGML application associated with the document or book.
<code>attrDisplayNew</code>	integer	Attribute display option for new elements. In other words the <code>attributeDisplay</code> attribute (on page 82) of every new structural element you create in this book or document will have this value. For list of possible values see "Options for displaying element attributes in structural view." on page 140..
<code>attrEditNew</code>	integer	Specifies whether and when the dialog box of attribute editor appears for new elements. For list of possible values see "Options for editing attributes of new structural elements." on page 141..
<code>borders</code>	integer	True if the book or document has normal borders in viewer mode; False if the border buttons are suppressed.
<code>catalogDisplay</code>	integer	Determines what elements will show in the element catalog window. For list of possible values see "Element catalog display options (what elements are listed)." on page 142.

Attribute	Type of value	Meaning
collate	integer	True if collated print is enabled (complete copy prints before another one). False if all copies of a page are printed together (and the print job is done more quickly).
customElements	list of strings	List of tags to display when catalogDisplay attribute (on page 21) is set to CustomCatalog.
elementCatalog	\N	List of elements in element catalog.
elementSelection	\N	The currently selected element range in structured book or document.
emulsionDirection	integer	Direction of print emulsion: EmulsionUp, EmulsionDown.
firstElemDefinition	FM object	First element definition in book or document (ElementDef object).
firstFormatChange	FM object	First format change list in the list of format change lists in the book or document (FormatChange object).
forbiddenFCodes	list of integers	F-codes that can't be executed in the book or document in viewer mode.
hasMenu	integer	True if the book or document has a window menu bar in viewer mode.
hasPopup	integer	True if the book or document window pop-up menu is available.
numCopies	integer	Number of copies to print.
paperHeight	integer	Height of paper to print to.
paperWidth	integer	Width of paper to print to.
path	string	Pathname of the book or document.
pdfBookmark	integer	True if bookmarks are to be generated when saving as PDF.

Attribute	Type of value	Meaning
pdfBookmarkElems	list of strings	List of element tags and context labels to include in bookmarks.
pdfDefaultsChanged	integer	True if the default heuristics for determining the paragraph level are disabled.
pdfGenerate	integer	True if special PDF data is to be generated. To generate it, you must set other print properties as follows: printToFile attribute (on page 24): True, printThumbnails attribute (on page 78): False, separations attribute (on page 24): False, print-BlankPages attribute (on page 23): True, reversePrint attribute (on page 24): False, numCopies attribute (on page 22): 1, printOdd attribute (on page 24): True, printEven attribute (on page 24): True, printScale attribute (on page 24): 100%.
pdfInfo	list of strings	An even number of strings that build PDF document info dictionary. Even indexed ones specify entry names and can be up to 126 characters long. Odd indexed ones contain entry content and may be up to 32765 characters long. You can use Unicode characters, but coded as ASCII strings (&#x....).
pdfNoThreads	integer	True if you do not want article threads in the resulting PDF.
pdfThreadColumns	integer	True if you want separate article threads for each column, False if you want separate article threads for each text frame. See also pdfNoThreads attribute (on page 23).
pdfUseBookmarkElems	integer	True if elements rather than paragraphs are used for bookmarks. Applies only to structured books and documents.
printBlankPages	integer	True if pageRounding attribute (on page 76) allows empty pages at the end of document(s) and these should be printed.

Attribute	Type of value	Meaning
printer	string	Name of printer. Setting this attribute on Windows has no effect. Once set, you can return to the default printer by setting the name to zero.
printEven	integer	True if even-numbered pages are to be printed.
printFile	string	Name of file to print to. Once set, you can return to the default file-name by setting to zero. Setting this property on the Macintosh has no effect.
printImaging	integer	Type of print imaging: ImagingPositive, ImagingNegative.
printLowRes	integer	True if printing in low resolution.
printOdd	integer	True if odd-numbered pages are to be printed.
printScale	integer	Scale factor (0% to 100%).
printToFile	integer	True if printing to file only.
registrationMarks	integer	True if registration marks are to be printed.
reversePrint	integer	True if reverse printing order is enabled.
separateInclusions	integer	True if inclusions are listed separately in the element catalog.
separations	integer	True if printing in separated color components is enabled.
skipBlankSeparations	integer	True if blank color separations are not to be printed.
status	string	String that appears in the status bar of the book or document. This attribute is effectively write-only. If you set it to a non-empty string, it remains until you set it to something else or an empty string. In the latter case FrameMaker will update it to normal status.

Attribute	Type of value	Meaning
tagsInBookmarks	integer	True if the paragraph tag is added before the paragraph text in each bookmark.
targetsMarked	integer	True if bookmark targets are already marked (by their markedDestination attribute (on page 102) and markedDestination attribute (on page 83)).
useInitialStructure	integer	True if FrameMaker+SGML inserts initial structure for new elements.

4.1.2 Book, document and session

Table 2: Book, Document, Session have the following attributes.

Attribute	Type of value	Meaning
iconified	integer	True if the book, document or FrameMaker product window is iconified (UNIX and Windows only).
inFront	integer	True if the book or document window is in front of other windows in the FrameMaker product session or the session is in front of other application windows. Can be used to bring a window to the front or back. However, reading this property on Unix is not always reliable because the XServer doesn't necessarily update this attribute immediately.
onScreen	integer	True if the book, document or FrameMaker product window is visible on the screen. Note that this attribute is always True for books, and setting it to False has no effect, as for the product window it is supported on UNIX only.

4.1.3 Book, document, session and dialog box

Table 3: Book, Document, DlgBox, Session have the following attributes.

Attribute	Type of value	Meaning
label	string	The title in the book, document or FrameMaker product window or dialog box title bar (for the product window on UNIX and Windows only).
screenHeight	integer	Height of the book, dialog box, document or FrameMaker product window in pixels (for the session on UNIX and Windows only).
screenWidth	integer	Width of the book, dialog box, document or FrameMaker product window in pixels (for the session on UNIX and Windows only).
screenX	integer	The offset of the book, dialog box, document or FrameMaker product window in pixels from the left side of the screen (for book, dialog box and document on Windows from the left of the FrameMaker product application window, for session on UNIX and Windows only). If you set a value that would result in the window being off the screen, that value is ignored and the old value is retained.
screenY	integer	The offset of the book, dialog box, document or FrameMaker product window in pixels from the top of the screen (for book, dialog box and document on Windows from the top of the FrameMaker product application window, for session on UNIX and Windows only). If you set a value that would result in the window being off the screen, that value is ignored and the old value is retained.

4.1.4 Book component and document

Table 4: Component, Document have the following attributes.

Attribute	Type of value	Meaning
chapNumber	integer	If the chapNumbering attribute (on page 27) is set to Restart-Num, this is the chapter number.
chapNumbering	integer	Option how the chapter number is to be computed. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
chapNumStyle	integer	Chapter numbering style. For list of possible values see "Styles available for autonumbering." on page 160.
chapNumText	string	If chapNumStyle attribute (on page 27) is set to NumText, use this text instead of number.
firstFootnoteNum	integer	Number of the first footnote when the fnoteNumbering attribute (on page 27) is set to RestartNum.
firstPageNum	integer	Page number of first page, when the pageNumbering attribute (on page 28) is set to RestartNum.
fnoteNumbering	integer	Option how the footnote numbers are to be computed. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
footnoteChars	string	Characters for custom footnote numbers.
footnoteNumStyle	integer	Footnote numbering style. For list of possible values see "Styles available for footnote autonumbering." on page 148.

Attribute	Type of value	Meaning
pageNumbering	integer	Option how the page numbers are to be computed. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
pageNumStyle	integer	Page numbering style. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
paraNumbering	integer	Paragraph numbering for the book component. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
tableFnoteChars	string	Characters for table footnote custom numbers.
tableFnoteStyle	integer	Numbering style for table footnotes. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
volumeNumber	integer	If the volumeNumbering attribute (on page 28) is set to RestartNum, this is the volume number.
volumeNumbering	integer	Option how the volume number is to be computed. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145.
volumeNumStyle	integer	Volume numbering style. For list of possible values see "Styles available for autonumbering." on page 160.
volumeNumText	string	If volumeNumStyle attribute (on page 28) is set to NumText, use this text instead of number.

4.1.5 Character related objects

Table 5: FormatChange, Paragraph, ParaFormat, CharFormat, Document, TxLocation, TxRange have the following attributes.

Attribute	Type of value	Meaning
capitalization	integer	Type of capitalization to use. For list of possible values see "When displaying letters there are these possibilities." on page 141.
changeBar	integer	True if change bars are enabled.
color	FM object	The spot color (Color object).
combinedFont	FM object	Combined font definition.
fontAngle	integer	Font angle - an index into the list of font angles provided by the session's fontAngleNames attribute (on page 109).
fontFamily	integer	Font family - an index into the list of font families provided by the session's fontFamilyNames attribute (on page 110).
fontSize	integer	Font size (between 2 and 400 pt).
fontVariation	integer	Font variation - an index into the list of font variations provided by the session's fontVariations attribute (on page 110).
fontWeight	integer	Font weight - an index into the array of font weights provided by the session's fontWeights attribute (on page 110).
kernX	integer	Horizontal kern value for manual kerning expressed as a percentage of an em. A positive value moves characters right and a negative value moves them left.
kernY	integer	Vertical kern value for manual kerning expressed as a percentage of an em. A positive value moves characters up and a negative value moves them down.

Attribute	Type of value	Meaning
outline	integer	True if outline style is enabled (Macintosh only).
overline	integer	True if overline style is enabled.
pairKern	integer	True if pair kerning is enabled.
position	integer	Position of characters relative to baseline of text: PositionNormal, PositionSubscript, PositionSuperscript.
shadow	integer	True if shadow style is enabled (Macintosh only).
spread	integer	Character spread in %.
stretch	integer	Character stretch (set width) expressed as a percentage of normal stretch for the font (-100% to 1000%).
strikeThrough	integer	True if strikethrough style is enabled.
underlining	integer	Type of underlining: NoUnderline, SingleUnderlineText, DoubleUnderlineText, NumericUnderlineText.

4.1.6 Graphic objects

Table 6: AFrame, Arc, Ellipse, Expression, UFrame, GInset, GLine, RoundRect, Polygon, Polyline, Rectangle, TxFrame, TxLine have the following attributes.

Attribute	Type of value	Meaning
angle	integer	Angle of object's rotation. Some objects can be rotated in multiplications of 90 degrees only.
arrowAngle	integer	Arrowhead base angle in degrees.

Attribute	Type of value	Meaning
arrowLength	integer	Arrowhead length (always rounded down to the nearest 1/256 point).
arrowScaleFactor	integer	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>scaleArrow</code> attribute (on page 32) is False.
arrowTipAngle	integer	Arrowhead tip angle in degrees.
arrowType	integer	Arrowhead style: <code>ArrowStick</code> , <code>ArrowHollow</code> , <code>Arrow-Filled</code> .
borderWidth	integer	Border width (0.015 pt to 360 pt).
cantBeSelected	integer	True if the graphic object can't be selected.
color	FM object	The spot color (<code>Color</code> object).
dashStyle	list of integers	Dash style. ???
fillPattern	integer	The fill pattern (numbers between 0 and 15). For list of possible values see "Fill patterns" on page 138.
frameParent	FM object	Frame containing the graphic object (<code>AFrame</code> or <code>UFrame</code> object).
hasArrow	integer	True if there is an arrowhead at end of line.
height	integer	Height of the object (0.125 pt to 3600 pt).
lineEndType	integer	Type of line end: <code>CapStyleButt</code> , <code>CapStyleRound</code> , <code>CapStyleProjecting</code> .
locX	integer	Distance from the left side of the parent frame (-216 inches to 216 inches). If the graphic object is an anchored frame, the distance is calculated from the left side of the page frame and is read-only.

Attribute	Type of value	Meaning
locY	integer	Distance from the top of the parent frame (-216 inches to 216 inches). If the graphic object is an anchored frame, the distance is calculated from the top of the page frame and is read-only.
next	FM object	Next graphic object in the document.
nextSelected	FM object	Next selected graphic object in document.
overprint	integer	Overprint settings for the object: PrintObjectOver, PrintObjectKO, PrintObjectColor.
pattern	integer	The fill pattern (numbers between 0 and 15). For list of possible values see "Fill patterns" on page 138.
scaleArrow	integer	True if arrowhead is scaled as the line width changes.
selected	integer	True if the graphic object is selected.
startArrow	integer	Arrowhead at beginning of line (True if enabled).
tintPercentage	integer	The tint percentage.
unique	integer	Persistent unique identifier of the object.
width	integer	Width of object (0.125 pt to 3600 pt).

4.1.7 Unanchored graphic objects

Table 7: Arc, Ellipse, Expression, UFrame, GInset, GLine, RoundRect, Polygon, Polyline, Rectangle, TxFrame, TxLine have the following attributes.

Attribute	Type of value	Meaning
groupParent	FM object	Group the object is in (Group object). Frames do not have this attribute.
nextInFrame	FM object	Next graphic object in the frame.
nextInGroup	FM object	Next graphic object in the group.
prevInFrame	FM object	Previous graphic object in the frame.
prevInGroup	FM object	Previous graphic object in the group.
runAround	integer	Specifies whether text can flow around the object and, if so, whether the text follows the contour of the object or a box shape surrounding the object: DoNotRunAround, RunAroundContour, RunAroundBox.
runAroundGap	integer	If the object is to be ran around, the width of the gap.

4.1.8 Menu related objects

Table 8: Command, Menu, Separator have the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the menu or menu item is enabled or False if it is disabled (dimmed).

Attribute	Type of value	Meaning
label	string	The label the user sees on a menu or command. The label for menu item separators is read-only; it is always ---.
name	string	The command, menu, or menu item separator name.
nextItemInMenu	FM object	The next menu item, menu, or separator within the menu.
nextItemInSession	FM object	The next menu item, menu, or separator in the list of menu items, menus, and separators in the session.
prevItem	FM object	The previous menu item, menu, or separator in the menu.

4.1.9 Paragraph related objects

Table 9: Paragraph, ParaFormat, FormatChange have the following attributes.

Attribute	Type of value	Meaning
adjHyphens	integer	Number of allowable adjacent hyphens.
aLetterSpaceMax	integer	Maximum asian letter space.
aLetterSpaceMin	integer	Minimum asian letter space.
aLetterSpaceOpt	integer	Optimum asian letter space.
alignment	integer	Horizontal alignment of the paragraph: ParaAlignLeft, ParaAlignRight, ParaAlignCenter, ParaAlignJustify.
aRomanSpaceMax	integer	Maximum asian-roman space.
aRomanSpaceMin	integer	Minimum asian-roman space.

Attribute	Type of value	Meaning
aRomanSpaceOpt	integer	Optimum asian-roman space.
autoNum	integer	True if autonumbering is enabled.
autoNumCharFormat	string	Character format for the automatic numbering string specified by autoNumFormat attribute (on page 35).
autoNumFormat	string	Format string for automatic numbering (for example, <n>.<n+>).
blockLines	integer	The lowest number of lines to keep together when breaking paragraph around column boundaries..
cellMarginBottom	integer	Space added to default bottom margin of table cell.
cellMarginLeft	integer	Space added to default left margin of table cell.
cellMarginRight	integer	Space added to default right margin of table cell.
cellMarginTop	integer	Space added to default top margin of table cell.
firstIndent	integer	Left margin of first line, measured from left side of current text column (0 cm to 100 cm).
frameAbove	string	Name of reference frame to put above the paragraph.
frameBelow	string	Name of reference frame to put below paragraph.
hyphenate	integer	True if automatic hyphenation is enabled.
hyphMinPrefix	integer	Minimal number of letters that must precede a hyphen.
hyphMinSuffix	integer	Minimal number of letters that must follow a hyphen.
hyphMinWord	integer	Minimal length of a hyphenated word.
keepWithNext	integer	True if the paragraph should be kept together with the next one.

Attribute	Type of value	Meaning
keepWithPrev	integer	True if the paragraph should be kept together with the preceding one.
language	integer	What language to use for hyphenation and spell-checking: Lang-Brazilian, LangBritish, LangFrenchCanadian, Lang-Catalan, LangDanish, LangDutch, LangEnglish, LangFinnish, LangFrench, LangGerman, LangItalian, LangNone, LangNorwegian, LangNyorsk, LangPortuguese, LangSpanish, LangSwedish, LangGermanSwiss, LangJapanese, LangChineseTraditional, LangChineseSimplified, LangKorean.
leading	integer	Space below each line in the paragraph.
leftIndent	integer	Left margin, measured from left side of current text column (0 cm to 100 cm).
maxWordSpacing	integer	Maximum space between words (percentage of an em space in current font).
minWordSpacing	integer	Minimum space between words (percentage of an em space in current font).
numAtEnd	integer	True if the paragraph number is placed at end and False if it is at begin of the paragraph.
placement	integer	Paragraph placement. For list of possible values see "Horizontal placement options for paragraphs." on page 164.
rightIndent	integer	Right margin, measured from right side of current text column.
runInSeparator	string	String to place between run-in head and the paragraph containing it.
spaceAbove	integer	Minimum space to left above the paragraph.

Attribute	Type of value	Meaning
<code>spaceBelow</code>	integer	Minimum space to left below the paragraph.
<code>start</code>	integer	Vertical placement of paragraph. For list of possible values see "Vertical placement options for paragraphs." on page 164.
<code>tabulators</code>	<code>\N</code>	List of tabulator descriptions.
<code>verticalAlign</code>	integer	Vertical alignment of a paragraph when first in a cell: <code>VerticalAlignTop</code> , <code>VerticalAlignMiddle</code> , <code>VerticalAlignBottom</code> .
<code>wordSpaceOpt</code>	integer	Optimum space between words (percentage of an em space in current font).
<code>wordSpacing</code>	integer	True if variable word spacing is enabled.
<code>yakumonoType</code>	integer	Yakumono rules to handle punctuation characters: <code>FloatingYakumono</code> , <code>MonospacedYakumono</code> , <code>FixedYakumono</code> .

4.1.10 Referenced text objects

Table 10: `TxInset`, `TbInset`, `FInset`, `PlugInset` have the following attributes.

Attribute	Type of value	Meaning
<code>autoupdate</code>	integer	True if the inset is updated automatically. It has no effect if the document's <code>dontUpdateTextInsets</code> attribute (on page 66) is set to True.
<code>file</code>	string	Pathname of the source file.

Attribute	Type of value	Meaning
<code>importHint</code>	string	Record identifying the filter used to import the text. The FrameMaker product uses this record to find the filter to use when updating the inset. For a more in-depth description of the syntax of this string, you should read the documentation available with FDK.
<code>lastModification</code>	string	Modification date of the text inset's source file.
<code>lastUpdate</code>	integer	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
<code>locked</code>	integer	True if the inset is locked. To change an inset's contents, you must unlock it. Always relock an inset after you have finished changing its contents.
<code>name</code>	string	A name assigned to the inset by a plug-in. It is not automatically assigned by the FrameMaker product.
<code>next</code>	FM object	Next text inset in the document (PlugInset, TxInset, TbInset, or FInset object).
<code>range</code>	TxRange object	Text range occupied by the text inset in the document containing the inset.
<code>sourceMac</code>	integer	If the source file is a Macintosh edition, the ID of its <code>sect</code> and <code>alis</code> records.
<code>unique</code>	integer	Persistent unique identifier of the object.

4.1.11 Table and table format

Table 11: Table, TbFormat have the following attributes.

Attribute	Type of value	Meaning
alignment	integer	Horizontal placement of table: TableCentered, TableLeft, TableRight.
bodyRowRuling	FM object	Ruling applied to body rows specified by bodyRowRulingPeriod attribute (on page 39) (RulingFormat object).
bodyRowRulingPeriod	integer	The periodicity of the ruling specified by bodyRowRuling attribute (on page 39). For example, if set to 3, the ruling specified by bodyRowRuling attribute (on page 39) is applied to every third row.
bottomMargin	integer	Default bottom cell margin for the table.
bottomRuling	FM object	Ruling applied to the bottom of the table (RulingFormat object).
colRuling	FM object	Ruling applied to table columns specified by colRulingPeriod attribute (on page 39) (RulingFormat object).
colRulingPeriod	integer	Periodicity of the ruling specified by colRuling attribute (on page 39). For example, if set to 3, the ruling specified by colRuling attribute (on page 39) is applied to every third column.
coumnsShaded	integer	True if table is shaded by columns; False if it is shaded by rows.
firstBodyColor	FM object	First spot color for table body shading (Color object).
firstBodyFill	integer	First fill pattern for table body shading.
firstBodyPeriod	integer	Number of columns or body rows to which the first shading (specified by firstBodyColor attribute (on page 39) and firstBodyFill attribute (on page 39)) is applied.

Attribute	Type of value	Meaning
hfColor	FM object	Spot color for shading of table heading and footing.
hfFill	integer	Fill pattern for shading table heading and footing. For list of possible values see "Fill patterns" on page 138.
hfRowRuling	FM object	Ruling for the heading and footing rows (RulingFormat object).
hfSeparatorRuling	FM object	Ruling that separates table header, body and footer (RulingFormat object).
inCatalog	integer	True if the format is in the catalog.
lastBodyRuling	integer	True if bottom ruling of the table appears on last sheet only (RulingFormat object).
leftIndent	integer	Left indent for the table.
leftMargin	integer	Default left cell margin for the table.
leftRuling	FM object	Ruling for the left side of the table (RulingFormat object).
name	string	Name of table format.
nextBodyColor	FM object	Second spot color for table body shading (Color object).
nextBodyFill	integer	Second fill pattern for table body shading.
nextBodyPeriod	integer	Number of columns or body rows to which the second shading (specified by nextBodyColor attribute (on page 40) and nextBodyFill attribute (on page 40)) is applied.
numbering	integer	Direction of autonumbering for the table: AutonumByCols, AutonumByRows.
orphanRows	integer	Number of orphan rows.

Attribute	Type of value	Meaning
otherColRuling	FM object	Ruling applied to table columns not specified by the colRuling-Period attribute (on page 39).
otherRowRuling	FM object	Ruling applied to body rows not specified by bodyRowRuling-Period attribute (on page 39).
placement	integer	Placement of table: PlaceTblAnywhere, TblAtTopOfColumn, TblAtTopOfPage, TblAtTopOfLeftPage, TblAtTopOfRightPage, TblPlaceFloat.
rightIndent	integer	Right indent for table.
rightMargin	integer	Default right cell margin for the table.
rightRuling	FM object	Ruling for the right side of the table (RulingFormat object).
spaceAbove	integer	Vertical space above table.
spaceBelow	integer	Vertical space below table.
titleGap	integer	Gap between the title and top or bottom row of the table.
titlePosition	integer	Placement of the table title. For list of possible values see "Options where to display table title." on page 168.
topMargin	integer	Default top cell margin for the table.
topRuling	FM object	Ruling for the top of the table (RulingFormat object).

4.2 Specific attributes

4.2.1 Anchored frame

Table 12: AFrame has the following attributes. See also: Table 6 on page 30.

Attribute	Type of value	Meaning
alignment	integer	Type of alignment: <code>FrameAlignCenter</code> , <code>FrameAlignInside</code> , <code>FrameAlignOutside</code> , <code>FrameAlignLeft</code> , <code>FrameAlignRight</code> .
anchorLocation	<code>TxLocation</code> object	Location of the frame anchor.
anchorType	integer	General positioning of the frame relative to the anchor. For list of possible values see "Types of anchoring. The frame will appear on the same page as its (text) anchor globally positioned accordingly to this type." on page 139.
baselineOffset	integer	Vertical offset of the frame relative to the baseline of the anchor.
cropped	integer	True if content of the frame is cropped.
element	FM object	If the anchored frame is in a structured flow in a structured document, the <code>Element</code> object containing the anchored frame.
firstGraphic	FM object	First graphic object in the frame.
floating	integer	True if floating is enabled.
inTextFrame	FM object	Text frame in which anchored frame appears (<code>TxFram</code> object).
inTextObject	FM object	Column or text frame in which anchored frame appears (<code>TxColumn</code> if it fits into the column, otherwise <code>TxFram</code> object).

Attribute	Type of value	Meaning
lastGraphic	FM object	Last graphic object in the frame (frontmost object).
nextInFrame	FM object	Next anchored frame in the text frame (AFrame object).
prev	FM object	Previous anchored frame within the same text frame (AFrame object).
sideOffset	integer	Horizontal offset of the anchored frame from column or text frame.

4.2.2 Graphic arc

Table 13: Arc has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
startAngle	integer	Start angle (0 degree to 360 degree).
theta	integer	Arc angle length (-360 degrees to 360 degrees).

4.2.3 Body page

Table 14: BodyPage has the following attributes.

Attribute	Type of value	Meaning
height	integer	Height of page.
isRight	integer	True for right body pages and False for left body pages.

Attribute	Type of value	Meaning
master	integer	Type of master page background. For list of possible values see "Types of background used for body pages." on page 141.
masterPage	string	Name of master page for body page if the master attribute (on page 44) is set to CustomPageBG. It is an empty string if the master attribute (on page 44) is set to DefaultPageBG or NoPageBG.
next	FM object	Next body page (BodyPage object) in the document.
number	integer	Page number.
numberString	string	String representation of page number.
pageFrame	FM object	Page frame (UFrame object).
pointPageNum	integer	Number of point page. (BTW, what's a point page?)
prev	FM object	Previous body page (BodyPage object) in the document.
width	integer	Width of page.

4.2.4 Book

Table 15: Book has the following attributes. See also: Table 1 on page 21, Table 2 on page 25, Table 3 on page 26.

Attribute	Type of value	Meaning
dontTouchReferences	integer	False if the FrameMaker product updates cross-references when it opens the book.
firstComponent	FM object	First component in the book (Component object).

Attribute	Type of value	Meaning
firstSelectedCompo	FM object	First selected component in the book (Component object).
modified	integer	True if the book has been modified.
next	FM object	Next open book (Book object).
selected	integer	True if the book icon in the book window is selected.
showComponents	integer	Option how the book components will be displayed (what will be shown next to the component's icon). For list of possible values see "Options how to display components in a book." on page 166..
topElement	FM object	Highest-level element if the book is structured (Element object).
viewOnly	integer	True if the book is view-only.

4.2.5 Book component

Table 16: Component has the following attributes. See also: Table 4 on page 27.

Attribute	Type of value	Meaning
book	FM object	Book that contains the component (Book object).
createLinks	integer	True if hypertext links are automatically inserted into generated files.
displayText	string	If the showComponents attribute (on page 45) is set to AsText, this string is displayed instead of component's filename, or contents of its first paragraph, when this value is an empty string.
element	FM object	Highest level element represented by the component (Element object).

Attribute	Type of value	Meaning
elementsToExtract	list of strings	List of element tags that are used to set up a generatable file (for example, table of contents, list of figures, or list of tables).
formatImportTarget	integer	True if the book component is included in the list of components to be updated with imported formats or element definitions when the user or a plug-in imports formats or element definitions into the book containing it.
generatable	integer	True if book component is a generated file (type attribute (on page 47) is not set to NotGenerated).
next	FM object	Next component in the book file (Component object).
nextSelected	FM object	Next selected component in the book window (Component object).
pageSide	integer	Page side to start the component document on: StartSideFromFile, StartSideNext, StartSideLeft, StartSideRight.
path	string	Pathname of document that the component represents.
prev	FM object	Previous component in the book file (Component object).
printInclude	integer	True if the component document is included in list of book files to be printed.
selected	integer	True if the component is selected in the book window.
tableFnoteNumbering	integer	Option how the footnote numbers in tables are to be computed. For list of possible values see "What numbers will be given to numbered objects (such as paragraphs or pages) in a book component." on page 145. However, in this case only RestartNum (start at 1) and NumFromFile are appropriate.

Attribute	Type of value	Meaning
<code>tagsToExtract</code>	list of strings	List of paragraph tags or marker type names that are used to set up a generatable file (for example, table of contents, list of figures, standard index or index of authors).
<code>type</code>	integer	Type of book component. For list of possible values see "Types of book components." on page 145..
<code>unique</code>	integer	Persistent unique identifier of the object.
<code>updateInclude</code>	integer	True if the component appears in the scroll list of files to be generated by the generate/update command for the book.

4.2.6 Table cell

Table 17: `TbCell` has the following attributes.

Attribute	Type of value	Meaning
<code>angle</code>	integer	Angle of rotation (multiplication of 90 degrees).
<code>bottomRulingOverride</code>	FM object	Cell's bottom ruling (<code>RulingFormat</code> object).
<code>cellAbove</code>	FM object	Cell above current cell (<code>TbCell</code> object).
<code>cellBelow</code>	FM object	Cell below current cell (<code>TbCell</code> object).
<code>colNum</code>	integer	Cell's column number.
<code>contentHeight</code>	integer	The distance between the top of the cell and the baseline of the last line in the cell.
<code>defaultRulingBottom</code>	FM object	Cell's default bottom ruling (<code>RulingFormat</code> object).

Attribute	Type of value	Meaning
defaultRulingLeft	FM object	Cell's default left ruling (RulingFormat object).
defaultRulingRight	FM object	Cell's default right ruling (RulingFormat object).
defaultRulingTop	FM object	Cell's default top ruling (RulingFormat object).
element	FM object	If the cell is in a structured document, the Element object containing the cell.
fillOverride	integer	Fill pattern or zero if there is no override fill pattern. For list of possible values see "Fill patterns" on page 138.
firstPgf	FM object	First paragraph in the cell (Paragraph object).
inTextFrame	FM object	Text frame containing the cell (TxFrame object).
inTextObject	FM object	Text object containing the cell (TxColumn object).
lastPgf	FM object	Last paragraph in the cell (Paragraph object).
leftRulingOverride	FM object	Cell's left ruling (RulingFormat object).
nextInFrame	FM object	Next cell in the text frame (TbCell object).
nextInRow	FM object	Next cell in current row from left to right (TbCell object).
nextInTable	FM object	Next cell from left to right (TbCell object). If the cell is at the end of a row, the next cell is the first cell in the next row.
overflowed	integer	True if the text in the cell overflows.
overrideBottomRuling	integer	True if the cell's bottom ruling (specified by bottomRulingOverride attribute (on page 47)) overrides the default ruling specified by the table format.

Attribute	Type of value	Meaning
<code>overrideFill</code>	integer	True if the cell's fill pattern (specified by <code>fillOverride</code> attribute (on page 48)) overrides the default fill pattern specified by the table format.
<code>overrideLeftRuling</code>	integer	True if the cell's left ruling (specified by <code>leftRulingOverride</code> attribute (on page 48)) overrides the ruling specified by the table format.
<code>overrideRightRuling</code>	integer	True if the cell's right ruling (specified by <code>rightRulingOverride</code> attribute (on page 49)) overrides the ruling specified by the table format.
<code>overrideShading</code>	integer	True if the cell's shading (specified by <code>shadingOverride</code> attribute (on page 49)) overrides the default shading specified by the table format.
<code>overrideTopRuling</code>	integer	True if the cell's top ruling (specified by <code>topRulingOverride</code> attribute (on page 50)) overrides the default top ruling specified by the table format.
<code>prevInFrame</code>	FM object	Previous cell in the text frame (<code>TbCell</code> object).
<code>prevInRow</code>	FM object	Previous cell in current row (<code>TbCell</code> object).
<code>rightRulingOverride</code>	FM object	Cell's right ruling (<code>RulingFormat</code> object).
<code>row</code>	FM object	Row containing the cell (<code>TbRow</code> object).
<code>shadingOverride</code>	FM object	Spot color (<code>Color</code> object).
<code>shown</code>	integer	True if the cell is conditional and visible.
<code>straddled</code>	integer	True if the cell is in a straddle but not the first cell.
<code>straddledCols</code>	integer	Number of columns in the straddle if the cell is the first in a horizontal straddle, otherwise 1.

Attribute	Type of value	Meaning
straddledRows	integer	Number of rows in the straddle if the cell is the first in a vertical straddle, otherwise 1.
topRulingOverride	FM object	Cell's top ruling (RulingFormat object).
unique	integer	Persistent unique identifier of the object.

4.2.7 Character format

Table 18: CharFormat has the following attributes. See also: Table 5 on page 29.

Attribute	Type of value	Meaning
charTag	string	Name of the character format.
fontEncoding	string	Encoding of the font.
fontPlatform	string	Name that uniquely identifies the Roman component of a combined font on a specific platform.
fontPlatformName	string	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
fontPostScript	string	Name given to the Roman component of a combined font when it is sent to a PostScript printer.
fontPSName	string	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.

Attribute	Type of value	Meaning
language	integer	What language to use for hyphenation and spell-checking: Lang-Brazilian, LangBritish, LangFrenchCanadian, LangCatalan, LangDanish, LangDutch, LangEnglish, LangFinnish, LangFrench, LangGerman, LangItalian, LangNone, LangNorwegian, LangNyorsk, LangPortuguese, LangSpanish, LangSwedish, LangGermanSwiss, LangJapanese, LangChineseTraditional, LangChineseSimplified, LangKorean.
name	string	Name of the character format.
next	FM object	Next character format in document (CharFormat object).
useAngle	integer	True if fontAngle attribute (on page 29) overrides default; False if As Is setting used.
useCapitalization	integer	True if capitalization attribute (on page 29) overrides default; False if As Is setting used.
useChangeBar	integer	True if changeBar attribute (on page 29) overrides default; False if As Is setting used.
useColor	integer	True if color attribute (on page 29) overrides default; False if As Is setting used.
useFamily	integer	True if fontFamily attribute (on page 29) overrides default; False if As Is setting used.
useKernX	integer	True if kernX attribute (on page 29) overrides default; False if As Is setting used.
useKernY	integer	True if kernY attribute (on page 29) overrides default; False if As Is setting used.

Attribute	Type of value	Meaning
<code>useOutline</code>	integer	True if <code>outline</code> attribute (on page 30) overrides default (Macintosh only); False if As Is setting used.
<code>useOverline</code>	integer	True if <code>overline</code> attribute (on page 30) overrides default; False if As Is setting used.
<code>usePairKern</code>	integer	True if <code>pairKern</code> attribute (on page 30) overrides default; False if As Is setting used.
<code>usePosition</code>	integer	True if <code>position</code> attribute (on page 30) overrides default; False if As Is setting used.
<code>useShadow</code>	integer	True if <code>shadow</code> attribute (on page 30) overrides default (Macintosh only); False if As Is setting used.
<code>useSize</code>	integer	True if <code>fontSize</code> attribute (on page 29) overrides default; False if As Is setting used.
<code>useSpread</code>	integer	True if <code>spread</code> attribute (on page 30) overrides default, False if As Is setting is used.
<code>useStretch</code>	integer	True if <code>stretch</code> attribute (on page 30) overrides default, False if As Is setting is used.
<code>useStrikethrough</code>	integer	True if <code>strikeThrough</code> attribute (on page 30) overrides default; False if As Is setting used.
<code>useUnderlining</code>	integer	True if <code>underlining</code> attribute (on page 30) overrides default; False if As Is setting used.
<code>useVariation</code>	integer	True if <code>fontVariation</code> attribute (on page 29) overrides default; False if As Is setting used.
<code>useWeight</code>	integer	True if <code>fontWeight</code> attribute (on page 29) overrides default; False if As Is setting used.

4.2.8 Color

Table 19: Color has the following attributes.

Attribute	Type of value	Meaning
black	integer	Percentage of black in the color (0% to 100%).
cyan	integer	Percentage of cyan (0% to 100%).
family	string	Number of last page to print. Note that the value of <code>fluidFlow</code> attribute (on page 68) must be 0; you can't print a range of pages when a document is in fluid view.
inkName	string	Name of the color library pigment.
magenta	integer	Percentage of magenta (0% to 100%).
name	string	Name of the color.
next	FM object	Next color in document (Color object).
overprint	integer	Overprint setting for the color: <code>ColorPrintsOver</code> , <code>ColorKnocksOut</code> .
printControl	integer	How to process the color when printing: <code>PrintSpot</code> , <code>PrintProcess</code> , <code>DoNotPrint</code> .
reserved	integer	Color reservation status (some colors are reserved by FrameMaker products): <code>ColorNotReserved</code> , <code>Cyan</code> , <code>Magenta</code> , <code>Yellow</code> , <code>Black</code> , <code>White</code> , <code>Red</code> , <code>Green</code> , <code>Blue</code> .
tintage	integer	The tint percentage (0% to 100%) or <code>ColorNotTinted</code> if the color is not a tint. Specifies the percentage of the <code>tintBase</code> attribute (on page 54) to use for tinting.

Attribute	Type of value	Meaning
tintBase	FM object	Color from which the tint is derived (Color object), or NoBaseColor if the color is not a tint.
viewControl	integer	A 12-bit number for spot color views. The least significant 2 bits are View 1, the next 2 bits are View 2, and so on.
yellow	integer	Percentage of yellow (0% to 100%).

4.2.9 Combined font definition

Table 20: CombiFont has the following attributes.

Attribute	Type of value	Meaning
baseFamily	integer	Asian font family - an index into the array of font families provided by the session's fontFamilyNames attribute (on page 110).
family	integer	Font family (specifies index into the array of font families provided by the session's fontFamilyNames attribute (on page 110)).
fontEncoding	string	Encoding of the combined font based on the baseFamily attribute (on page 54).
name	string	Name of the combined font.
next	integer	Next combined font definition instance in the document (CombiFont object).
shift	integer	Baseline offset of Roman text expressed as a percentage of base font size (metric 1% to 1000%).

Attribute	Type of value	Meaning
size	integer	Scaling factor for Roman text expressed as a percentage of base font size (metric 1% to 1000%).
userString	string	A string to which private data can be stored.

4.2.10 Command

Table 21: Command has the following attributes. See also: Table 8 on page 33.

Attribute	Type of value	Meaning
alterCommand	FM object	If shiftVariability attribute (on page 57) is set to HasShiftRelative, the command to use when the user holds down the Shift key. If shiftVariability attribute (on page 57) is set to HasUnshiftRelative, the command to use when the user isn't holding down the Shift key.
checkMarkAble	integer	True if the menu item can have a check mark. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.
checkOn	integer	True if the menu item can have a check mark and the check mark is on. If the menu item is defined by the FrameMaker product, you can get this attribute, but not set it.
enabledWhen	integer	The context in which the command is enabled. For a list of the constants that this field can specify, see the table below. If the menu item is defined by the FrameMaker product, you can get this property, but not set it.

Attribute	Type of value	Meaning
<code>expandomaticParent</code>	FM object	If the menu item is an <code>expandomatic</code> menu item, its virtual parent (Command object).
<code>fcode</code>	integer	An f-code that the <code>FrameMaker</code> product executes when the user chooses the menu item or uses the keyboard shortcut.
<code>fcodes</code>	list of integers	The list of f-codes that the <code>FrameMaker</code> product executes when the user chooses the menu item or uses the keyboard shortcut (the macro). Normally, the first f-code in the list is the same as the f-code specified by the <code>fcode</code> attribute (on page 56).
<code>helpLink</code>	string	The hypertext link to call when the user requests context-sensitive help for the command. If you set this property, specify the destination file and an optional page number or linkname. For example, specify <code>foo.fm:lastpage</code> . Do not specify hypertext commands such as <code>gotopage</code> . The <code>FrameMaker</code> product automatically prepends the appropriate hypertext command to the <code>helpLink</code> attribute (on page 56) when the user requests context-sensitive help. If the destination file is not in the client directory, the <code>FrameMaker</code> product looks for it in the <code>FrameMaker</code> product help directory. This attribute is valid only for commands created by clients. It is not valid for commands created directly by <code>FrameMaker</code> products.
<code>labels</code>	list of strings	If the command is a menu item, the list of labels it can have in different contexts. If the menu item has only one label in all contexts, it specifies only the string for that label. If the menu item has different labels in different contexts, it specifies pairs of strings with the format <code>Context,Label</code> where <code>Label</code> specifies the menu item label and <code>Context</code> specifies the context in which the label appears on the menu.

Attribute	Type of value	Meaning
mode	integer	The mode in which keyboard shortcuts are active: Short-cutAllModes, ShortcutMathMode, ShortcutNonmath-Mode.
next	FM object	The next command in the list of commands in the session.
shiftVariability	integer	Specifies whether a command has an accompanying shift or unshift command: HasShiftRelative, HasUnshiftRelative, HasNoRelative.
shortcutLabel	string	The keyboard shortcut string that appears on the menu. This string need not be one of the actual shortcuts specified by shortcuts attribute (on page 57).
shortcuts	list of strings	The list of keyboard shortcuts the user can use to execute the command. To add a shortcut, append it to the list. FrameMaker ignores deletions from the list.
type	integer	Type of command. For list of possible values see "Types of commands." on page 145.

4.2.11 Condition format

Table 22: CondFormat has the following attributes.

Attribute	Type of value	Meaning
conditionShown	integer	True if the text marked by the condition is shown. See also the showAll attribute (on page 79) of the Document object.
name	string	Name of the condition format.

Attribute	Type of value	Meaning
next	FM object	Next condition format in document (CondFormat object).
separationOverride	FM object	Color separation format override (Color object).
styleOverride	integer	Style condition indicators for conditional text: ChangeBar, DoubleUnderline, NoOverride, Overline, SingleUnderline, Strikethrough, NumericUnderline, NumericAndBar.
useSepOverride	integer	True if color specified by separationOverride attribute (on page 58) is used instead of default.

4.2.12 Dialog resource

Table 23: DlgResource has the following attributes.

Attribute	Type of value	Meaning
helpLink	string	The hypertext link to call when the user requests context-sensitive help for the dialog box. If you set this property, specify the destination file and an optional page number or linkname. For example, specify foo.fm:lastpage. Do not specify hypertext commands such as gotopage. The FrameMaker product automatically prepends the appropriate hypertext command to it when the user requests context-sensitive help. If the destination file is not in the plug-in directory, the FrameMaker product looks for it in the FrameMaker product help directory. If your plug-in has not defined a help link for the dialog box, this string is the default help link to the help main menu. The FrameMaker product includes dialog boxes that are shared by multiple commands. The string for such dialog boxes is prepended with a Shared db: statement.
numItems	integer	The number of items in the dialog box.

4.2.13 Dialog box

Table 24: DlgBox has the following attributes. See also: Table 3 on page 26.

Attribute	Type of value	Meaning
visible	integer	True if the box is visible.

4.2.14 Dialog button

Table 25: DlgButton has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the button is enabled; False if it is disabled (dimmed).
label	string	The label that appears on the button.
state	integer	State of the button. For list of possible values see "States of dialog controls." on page 146.
visible	integer	True if the button is visible.

4.2.15 Check box

Table 26: DlgCheckBox has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the checkbox is enabled; False if it is disabled (dimmed).
label	string	The label that appears next to the checkbox.
state	integer	State of the checkbox. For list of possible values see "States of dialog controls." on page 146.
visible	integer	True if the checkbox is visible.

4.2.16 Edit box

Table 27: `DlgEditBox` has the following attributes.

Attribute	Type of value	Meaning
<code>enabled</code>	integer	True if the text box is enabled; False if it is disabled (dimmed).
<code>text</code>	string	The text in the text box.
<code>visible</code>	integer	True if the text box is visible.

4.2.17 Dialog image

Table 28: `DlgImage` has the following attributes.

Attribute	Type of value	Meaning
<code>enabled</code>	integer	True if the image pop-up menu is enabled; False if it is disabled (dimmed).
<code>labels</code>	list of strings	The list of settings displayed in the image pop-up menu. The first string in the list of strings is the pop-up menu's label. On UNIX platforms, this label appears when the user clicks on the pop-up menu. It does not appear on the Macintosh or Windows platforms.
<code>state</code>	integer	The index (in the list specified by <code>labels</code> attribute (on page 61)) of the chosen setting. If no setting is chosen, it is -1.
<code>visible</code>	integer	True if the image pop-up menu is visible.

4.2.18 Dialog label

Table 29: DlgLabel has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the label is enabled; False if it is disabled (dimmed).
label	string	The string displayed by the label.
visible	integer	True if the label is visible.

4.2.19 Pop-up menu

Table 30: DlgPopUp has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the pop-up menu is enabled; False if it is disabled (dimmed).
labels	list of strings	The list of settings displayed in the pop-up menu. The first string in the list of strings is the pop-up menu's label. On UNIX platforms, this label appears when the user clicks on the pop-up menu. It does not appear on the Macintosh or Windows platforms.
state	integer	The index (in the list specified by labels attribute (on page 62)) of the chosen setting. If no setting is chosen, it is -1.
visible	integer	True if the pop-up menu is visible.

4.2.20 Radio button

Table 31: `DlgRadioButton` has the following attributes.

Attribute	Type of value	Meaning
<code>enabled</code>	integer	True if the radio button is enabled; False if it is disabled (dimmed).
<code>label</code>	string	The label that appears next to the radio button.
<code>state</code>	integer	State of the radio button. For list of possible values see "States of dialog controls." on page 146. Setting this attribute to <code>InactiveState</code> has no effect. To turn a radio button off, you must turn on another of the buttons in the button set.
<code>visible</code>	integer	True if the radio button is visible.

4.2.21 Scroll bar

Table 32: `DlgScrollBar` has the following attributes.

Attribute	Type of value	Meaning
<code>enabled</code>	integer	True if the scroll bar is enabled; False if it is disabled (dimmed).
<code>maxValue</code>	integer	The largest value to which the user can drag the scroll bar.
<code>pageIncrement</code>	integer	The amount that the scroll bar's thumb box moves when the user clicks on either side of it in scroll bar. For example, if set to 10, the scroll box moves 10 increments to the left when the user clicks to the left of the thumb box.

Attribute	Type of value	Meaning
size	integer	The scroll bar width if the scroll bar is horizontal or the scroll bar height if the scroll bar is vertical.
state	integer	The value of the scroll bar.
valueMin	integer	The smallest value to which the user can drag the scroll bar.
visible	integer	True if the scroll bar is visible.

4.2.22 Scroll box

Table 33: DlgScrollBox has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the scroll list is enabled; False if it is disabled (dimmed).
firstDisplayedItem	integer	The index (in the list specified by labels attribute (on page 64)) of the item that appears at the top of the scroll list.
labels	list of strings	The list of items in the scroll list.
numLines	integer	The number of items displayed in the scroll list.
state	integer	The index (in the list specified by labels attribute (on page 64)) of the chosen setting. If no setting is chosen, it is -1.
visible	integer	True if the scroll list is visible.

4.2.23 Three-state checkbox

Table 34: DlgTriBox has the following attributes.

Attribute	Type of value	Meaning
enabled	integer	True if the tribox is enabled; False if it is disabled (dimmed).
label	string	The label that appears next to the three-state box.
state	integer	State of the three-state box. For list of possible values see "States of dialog controls." on page 146.
visible	integer	True if the three state box is visible.

4.2.24 Document

Table 35: Document has the following attributes. See also: Table 1 on page 21, Table 2 on page 25, Table 3 on page 26, Table 4 on page 27, Table 5 on page 29.

Attribute	Type of value	Meaning
abovePgfMax	integer	Maximum space above paragraph allowed in the document.
abovePgfMin	integer	Minimum space above paragraph allowed in the document.
autoChangeBars	integer	True if automatic change bars are enabled.
belowPgfMax	integer	Maximum space below paragraph allowed in the document.
belowPgfMin	integer	Minimum space below paragraph allowed in the document.
bottomMargin	integer	Bottom page margin.

Attribute	Type of value	Meaning
changeBarColor	FM object	The spot color of change bars (Color object).
changeBarDistance	integer	Distance between change bar and text column.
changeBarPosition	integer	Position of change bars. For list of possible values see "Options for positioning change bars." on page 142.
changeBarWidth	integer	Width of change bars.
charTag	string	Name of character format applied to current point of insertion.
colorView	integer	Spot color separation view (0 to 5). 0 specifies View 1, 1 specifies View 2, and so on.
columnGap	integer	Size of gap between text columns.
conditions	list of integers	Conditions applied to the text at insertion point (list of CondFormat objects).
conditionShown	integer	True if condition at insertion point is shown.
currentPage	FM object	Current page (BodyPage, MasterPage or RefPage object).
dictionary	list of strings	List of words to accept when spell-checking the document.
displayUnit	integer	Equivalent of one unit in the current display units. For example, if points were selected for display units, this returns 65536.
dontUpdateTextInsets	integer	True if FrameMaker product doesn't automatically update text insets whenever opening the document.
dontUpdateXRefs	integer	True if FrameMaker product doesn't automatically update cross-references whenever printing the document.
doubleSided	integer	True if the document has two-sided page layout.

Attribute	Type of value	Meaning
downloadFonts	integer	For UNIX, the fonts to download to the printer when printing. Can be one of: DownloadNoFont (default), DownloadAllFonts, DownloadBut13Fonts, DownloadBut35Fonts.
elementBounds	integer	Element boundary display option. For list of possible values see "Options for displaying element bounds." on page 146.
firstBodyPage	FM object	First body page in the document (BodyPage object).
firstCharFmt	FM object	First character tag in the list of the document's character tags (CharFormat object).
firstColor	FM object	First color in the list of document's colors (Color object).
firstCombiFontDef	FM object	First combined font definition in the list of the document's combined font definitions (CombiFont object).
firstCondFmt	FM object	First condition tag in the list of the document's condition tags (CondFormat object).
firstFlow	FM object	First flow in the list of the document's flows (Flow object).
firstFootnote	FM object	First footnote in the list of the document's footnotes (Footnote object).
firstGraphic	FM object	First graphic object in the list of the document's graphic objects (object).
firstMarker	FM object	First marker in the list of the document's markers (Marker object).
firstMarkerType	FM object	First marker type in the list of the document's marker types (MarkerType object).
firstMasterPage	FM object	First master page in the document (MasterPage object).
firstPageVerso	integer	True if the document begins with a left page.

Attribute	Type of value	Meaning
<code>firstPgf</code>	FM object	First paragraph in the list of the document's paragraphs (Paragraph object).
<code>firstPgFmt</code>	FM object	First paragraph format in the list of the document's paragraph formats (ParaFormat object).
<code>firstRefPage</code>	FM object	First reference page in the document (RefPage object).
<code>firstRubi</code>	FM object	First rubi composite in the list of the document's rubi composites (Rubi object).
<code>firstRulingFmt</code>	FM object	First ruling format in the list of the document's ruling formats (RulingFormat object).
<code>firstSelectedGraphic</code>	FM object	First of selected graphic objects in the document (object).
<code>firstSelectedTi</code>	FM object	First of selected text insets in the document (PlugInset, TxInset, TbInset, or FInset object).
<code>firstTable</code>	FM object	First of tables in the document (Table object).
<code>firstTbFormat</code>	FM object	First of table formats in the document (TbFormat object).
<code>firstTi</code>	FM object	First of text insets in the document (PlugInset, TxInset, TbInset or FInset object).
<code>firstVar</code>	FM object	First of variables in the document (Variable object).
<code>firstVarFmt</code>	FM object	First of variable formats in the document (VarFormat object).
<code>firstXRef</code>	FM object	First of cross-references in the document (XRef object).
<code>firstXRefFmt</code>	FM object	First of cross-reference formats in the document (XRefFormat object).
<code>fluidFlow</code>	FM object	The flow to set to fluid view. To turn this off, set the value to 0.

Attribute	Type of value	Meaning
fontEncoding	string	Encoding of the font at insertion point.
fontPlatformName	string	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
fontPSName	string	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
fontSizeUnit	integer	Equivalent of one unit in the current unit for font size. Font size units can be either points or Q. If points, this returns 65536. If Q, this returns 47098.
footnoteFormat	string	Paragraph tag of document's footnotes.
footnoteMaxHeight	integer	Maximum height allowed for document footnotes (36 pt to 32767 pt).
footnoteNumPosition	integer	Placement of footnote number in footnote: FnNumSuper: superscript, FnNumBase: baseline, FnNumSub: subscript.
footnoteNumPrefix	string	Prefix to appear before footnote number in footnote.
footnoteNumSuffix	string	Suffix to appear after footnote number in footnote.
footnoteRefPosition	integer	Position of footnote reference in document text: FnNumSuper: Superscript, FnNumBase: Baseline, FnNumSub: Subscript.
footnoteRefPrefix	string	Prefix to appear before footnote number in document text.
footnoteRefSuffix	string	Suffix to appear after footnote number in document text.
forgetTouch	integer	False by default. Setting this to True allows to modify the document without the FrameMaker product setting modified attribute (on page 76) to True.

Attribute	Type of value	Meaning
<code>formatOverride</code>	integer	Specifies whether there are format overrides at the current insertion point. If text at the insertion point has a character format applied to it, it is only True if the text formatting overrides the character format. If text at insertion point does not have special character format applied to it, it is only True if the paragraph has formatting that overrides the catalog format.
<code>grayGraphics</code>	integer	True if graphics are not displayed, only gray boxes appear instead.
<code>gridUnit</code>	integer	Distance of grid lines.
<code>hiddenPage</code>	FM object	Hidden page in the document (HiddenPage object).
<code>hSpreadLarge</code>	integer	Horizontal spread (in %) in large mathematic expressions.
<code>hSpreadMedium</code>	integer	Horizontal spread (in %) in medium mathematic expressions.
<code>hSpreadSmall</code>	integer	Horizontal spread (in %) in small mathematic expressions.
<code>hypertextAlertTitle</code>	string	Read interface to the hypertext command parser. If <code>hypertext-CommandCode</code> attribute (on page 71) is <code>CmdAlertTitle</code> , the specified title for the alert box.
<code>hypertextClient</code>	string	Read interface to the hypertext command parser. For message commands, the name of the plug-in that receives the message.
<code>hypertextCommand</code>	string	Write interface to the hypertext command parser. Setting this value executes the parser, if <code>hypertextValidate</code> attribute (on page 72) is True. The command set to this attribute will be parsed and validated.

Attribute	Type of value	Meaning
<code>hypertextCommandCode</code>	integer	Read interface to the hypertext command parser. The FrameMaker hypertext command in <code>hypertextCommand</code> attribute (on page 70), as determined by the parser. For list of possible values see "Possible values to which <code>hypertextCommandCode</code> attribute (on page 71) can be set." on page 148.
<code>hypertextDestID</code>	integer	Read interface to the hypertext command parser. For links to objects, the unique identifier of the object in the target document.
<code>hypertextDestination</code>	integer	Read interface to the hypertext command parser. For link commands, the destination type in <code>hypertextCommand</code> attribute (on page 70), as determined by the parser. For list of possible values see "Possible values to which <code>hypertextDestination</code> attribute (on page 71) can be set." on page 150.
<code>hypertextDestType</code>	integer	Read interface to the hypertext command parser. For links to objects, the type of the object in the target document. For list of possible values see "Possible values to which <code>hypertextDestType</code> attribute (on page 71) can be set." on page 150.
<code>hypertextDIFFile</code>	string	Read interface to the hypertext command parser. For links to external files, the absolute path to the target file, expressed in platform independent syntax.
<code>hypertextError</code>	integer	Read interface to the hypertext command parser. Non-zero if there was a parse error. For list of possible values see "Possible error codes to which <code>hypertextError</code> attribute (on page 71) can be set." on page 150.
<code>hypertextErrorIndex</code>	integer	Read interface to the hypertext command parser. If there was a parse error, an index into the <code>hypertextTokens</code> attribute (on page 72) list.

Attribute	Type of value	Meaning
hypertextErrorMessage	string	Read interface to the hypertext command parser. The message <code>FrameMaker</code> generates for a parse error.
hypertextFlow	string	Read interface to the hypertext command parser. For popup and matrix commands, the name of the reference flow containing the popup or matrix list of commands.
hypertextLink	string	Read interface to the hypertext command parser. For links to named targets, either the value of a <code>newlink</code> command, or a keyword such as <code>FirstPage</code> or <code>LastPage</code> .
hypertextMatrixCols	integer	Read interface to the hypertext command parser. If <code>hypertext-CommandCode</code> attribute (on page 71) is <code>CmdMatrix</code> , the number of columns in the matrix.
hypertextMatrixRows	integer	Read interface to the hypertext command parser. If <code>hypertext-CommandCode</code> attribute (on page 71) is <code>CmdMatrix</code> , the number of rows in the matrix.
hypertextMessage	string	Read interface to the hypertext command parser. If <code>hypertext-CommandCode</code> attribute (on page 71) is <code>CmdAlert</code> , <code>CmdAlert-Title</code> , or <code>CmdMessage</code> , the specified message for the hypertext command.
hypertextPage	string	Read interface to the hypertext command parser. For links to pages, the page number.
hypertextTokens	\N	Read interface to the hypertext command parser. The value of <code>hypertextCommand</code> attribute (on page 70), parsed into individual tokens.
hypertextValidate	integer	Write interface to the hypertext command parser. If set to <code>True</code> , the next string sent to <code>hypertextCommand</code> attribute (on page 70) will be parsed and validated as hypertext command.

Attribute	Type of value	Meaning
hypertextValidateErr	integer	Read interface to the hypertext command parser. Non-zero if hypertextValidate attribute (on page 72) was true and there was a validation error. For list of possible values see "Possible error codes to which hypertextValidateErr attribute (on page 73) can be set." on page 151.
inheritedExclusions	list of strings	List of exclusions inherited when document is included in a structured book.
inheritedInclusions	list of strings	List of inclusions inherited when document is included in a structured book.
isHelp	integer	True if the document is the FrameMaker product's Help document.
isPalette	integer	True if the document is a view only palette.
lastBodyPage	FM object	Last body page in the document (BodyPage object).
lastMasterPage	FM object	Last master page (MasterPage object).
lastRefPage	FM object	Last reference page in the document (RefPage object).
leftMargin	integer	Left page margin.
leftMasterPage	FM object	Left master page (MasterPage object).
lineBreakAfter	string	Characters at which it is permissible to break lines.
mainFlow	FM object	Main flow in the document (Flow object).
markerNames	list of strings	List of names of marker types.
mathFunctionTag	string	Character format tag of equation font to apply to mathematic functions.
mathIntegralSizeLarge	integer	Point size of integral symbol in large equations (2 pt to 400 pt).

Attribute	Type of value	Meaning
mathIntegralSizeMed	integer	Point size of integral symbol in medium equations (2 pt to 400 pt).
mathIntegralSizeSmall	integer	Point size of integral symbol in small equations (2 pt to 400 pt).
mathLevel1SizeLarge	integer	Point size of level 1 expression in large equations (2 pt to 400 pt).
mathLevel1SizeMedium	integer	Point size of level 1 expression in medium equations (2 pt to 400 pt).
mathLevel1SizeSmall	integer	Point size of level 1 expression in small equations (2 pt to 400 pt).
mathLevel2SizeLarge	integer	Point size of level 2 expression in large equations (2 pt to 400 pt).
mathLevel2SizeMedium	integer	Point size of level 2 expression in medium equations (2 pt to 400 pt).
mathLevel2SizeSmall	integer	Point size of level 2 expression in small equations (2 pt to 400 pt).
mathLevel3SizeLarge	integer	Point size of level 3 expression in large equations (2 pt to 400 pt).
mathLevel3SizeMedium	integer	Point size of level 3 expression in medium equations (2 pt to 400 pt).
mathLevel3SizeSmall	integer	Point size of level 3 expression in small equations (2 pt to 400 pt).
mathNumberTag	string	Character format tag of equation font to apply to numbers in expressions.
mathSigmaSizeLarge	integer	Point size of sigma symbol in large equations (2 pt to 400 pt).
mathSigmaSizeMedium	integer	Point size of sigma symbol in medium equations (2 pt to 400 pt).
mathSigmaSizeSmall	integer	Point size of sigma symbol in small equations (2 pt to 400 pt).
mathStringTag	string	Character format tag of equation font to apply to strings in expressions.
mathSymbolTag	string	Character format tag of equation font to apply to symbols in expressions.

Attribute	Type of value	Meaning
mathVariableTag	string	Character format tag of equation font to apply to variables in expressions.
maxBottomMargin	integer	Maximum bottom margin allowed in the document.
maxFirstIndent	integer	Maximum first line indent allowed in the document.
maxFontSize	integer	Maximum font size allowed in the document.
maxLeading	integer	Maximum leading allowed in the document.
maxLeftIndent	integer	Maximum left indent allowed in the document.
maxLeftMargin	integer	Maximum left margin allowed in the document.
maxRightIndent	integer	Maximum right indent allowed in the document.
maxRightMargin	integer	Maximum right margin allowed in the document.
maxSpread	integer	Maximum spread value allowed in the document.
maxStretch	integer	Maximum character stretch (set width) expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
maxTab	integer	Maximum tab position allowed in the document.
maxTopMargin	integer	Maximum top margin allowed in the document.
menu	FM object	Menu bar of the document (Menu object).
minBottomMargin	integer	Minimum bottom margin allowed in the document.
minFirstIndent	integer	Minimum first indent allowed in the document.
minFontSize	integer	Minimum font size allowed in the document.
minLeading	integer	Minimum leading allowed in the document.

Attribute	Type of value	Meaning
<code>minLeftIndent</code>	integer	Minimum left indent allowed in the document.
<code>minLeftMargin</code>	integer	Minimum left margin allowed in the document.
<code>minRightIndent</code>	integer	Minimum right indent allowed in the document.
<code>minRightMargin</code>	integer	Minimum right margin allowed in the document.
<code>minSpread</code>	integer	Minimum spread value allowed in the document.
<code>minStretch</code>	integer	Minimum character stretch (set width) expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
<code>minTab</code>	integer	Minimum tab position allowed in the document.
<code>modified</code>	integer	True if document has been modified. While this property is read-only, you can modify a document without setting this attribute to True by setting <code>forgetTouch</code> attribute (on page 69) to True for the document before your client modifies it.
<code>next</code>	FM object	Next open document (Document object).
<code>numCols</code>	integer	Number of columns. ???
<code>openType</code>	integer	Type of document the file was opened as. For list of possible values see "Types of documents FrameMaker can open or save." on page 146.
<code>overhang</code>	integer	True if rubi is allowed to overhang.
<code>pageHeight</code>	integer	Height of the document's pages. Setting this property automatically sets the <code>height</code> attribute (on page 43) of all pages in the document.
<code>pageRounding</code>	integer	How to round page count. For list of possible values see "Options for page count rounding." on page 164.

Attribute	Type of value	Meaning
pageScroll	integer	Page scrolling: VariableScroll, HorizontalScroll, VerticalScroll, FacingScroll.
pageWidth	integer	Width of the document's pages. Setting this property automatically sets the width attribute (on page 44) of all body pages in the document.
pdfStructure	integer	If set to true a structured PDF will be created when saving as PDF. Structure is taken from the structureLevel attribute (on page 103) of ParaFormat objects.
pointPageNumStyle	integer	Point page numbering style. For list of possible values see "Styles available for autonumbering." on page 160.
printCols	integer	If printThumbnails attribute (on page 78) is True, the number of columns to print.
printEndPage	integer	Number of last page to print. Note that the value of fluidFlow attribute (on page 68) must be 0; you can't print a range of pages when a document is in fluid view.
printEndPageName	string	String representing number of last page to print if the numbering style is not set to arabic.
printEndPoint	integer	Number of last point page to print.
printManualFeed	integer	True if manual feed is enabled.
printRows	integer	If printThumbnails attribute (on page 78) is True, the number of rows to print.
printScope	integer	Print scope: PrintAll, PrintRange. Note that the value of fluidFlow attribute (on page 68) must be 0 (False) when printing ranges of pages: you can't print a range of pages when a document is in fluid view.

Attribute	Type of value	Meaning
<code>printStartPage</code>	integer	Number of first page to print. Note that the value of <code>fluidFlow</code> attribute (on page 68) must be 0; you can't print a range of pages when a document is in fluid view.
<code>printStartPageName</code>	string	String representing number of first page to print if the numbering style is not set to arabic.
<code>printStartPoint</code>	integer	Number of first point page to print.
<code>printThumbnails</code>	integer	True if printing of thumbnails is enabled.
<code>rightMargin</code>	integer	Right page margin.
<code>rightMasterPage</code>	FM object	Right master page (MasterPage object).
<code>rubiFixedSize</code>	integer	Fixed size for all rubi text (metric 2pts to 400pts). If this property and the <code>rubiSize</code> attribute (on page 78) property both have values, the most recently set property value is used.
<code>rubiLineAlignment</code>	integer	True if rubi and oyamoji text should be aligned at line boundaries.
<code>rubiNarrowSpaceKanji</code>	integer	Allowable values are: <code>RubiSpaceWide</code> , <code>RubiSpaceNarrow</code> , <code>RubiSpaceProportional</code> .
<code>rubiNarrowSpaceOther</code>	integer	Allowable values are: <code>RubiSpaceWide</code> , <code>RubiSpaceNarrow</code> , <code>RubiSpaceProportional</code> .
<code>rubiSize</code>	integer	Scaling factor for rubi text expressed as percentage of the current font size (metric 1% to 1000%). If this and the <code>rubiFixedSize</code> attribute (on page 78) both have values, the value most recently set is used.
<code>rubiWideSpaceKanji</code>	integer	Valid values are: <code>RubiSpaceWide</code> , <code>RubiSpaceNarrow</code> , <code>RubiSpaceProportional</code> .

Attribute	Type of value	Meaning
rubWideSpaceOther	integer	Valid values are: RubiSpaceWide, RubiSpaceNarrow, RubiSpaceProportional.
rulerUnit	integer	Measurement units to display on rulers.
saveType	integer	Type of document the file was saved as. For list of possible values see "Types of documents FrameMaker can open or save." on page 146.
select	integer	Specifies whether user can select text or graphics in the view only document. For list of possible values see "Selection options in view only mode." on page 170.
selectedTable	FM object	If any table cells are selected, the table containing them (Table object).
separationOverride	FM object	Custom color separation override at the insertion point (Color object).
showAll	integer	True if all conditions are to be displayed.
showConditions	integer	True if condition indicators (format overrides) are to be displayed.
smallCapsSize	integer	Scaling factor for small capitals expressed as percentage of the relevant font size (metric 1% to 1000%).
smallCapsStretch	integer	Character stretch (set width) for small caps expressed as a percentage of normal stretch for the font (-10% to 1000%).
smartQuotes	integer	True if smart quotes are enabled.
smartSpaces	integer	True if smart spaces are enabled.
snapAngle	integer	Angle of rotation for snap rotate.
snapGrid	integer	Spacing units of snap grid (0 to 32768 pt).

Attribute	Type of value	Meaning
styleOverride	integer	Style condition indicators for conditional text: DoubleUnderline, NoOverride, Overline, SingleUnderline, Strikethrough.
subScriptShift	integer	Baseline offset of subscripts expressed as percentage of current font size (1% to 1000%).
subScriptSize	integer	Scaling factor for subscripts expressed as percentage of current font size (1% to 1000%).
subScriptStretch	integer	Character stretch (set width) for subscripts expressed as a percentage of normal stretch for the font (-10% to 1000%).
superScriptShift	integer	Baseline offset of superscripts expressed as percentage of current font size (1% to 1000%).
superScriptSize	integer	Scaling factor for superscripts expressed as percentage of the current font size (1% to 1000%).
superScriptStretch	integer	Character stretch (set width) for superscripts expressed as a percentage of normal stretch for the font (-10% to 1000%).
symbolFonts	list of strings	List of math symbol fonts used in Equation Fonts dialog box.
tableFnoteFormat	string	Paragraph tag of table footnotes.
tableFnotePosition	integer	Placement of footnote number in footnote text: FnNumSuper: superscript, FnNumBase: baseline, FnNumSub: subscript.
tableFnotePrefix	string	Prefix appearing before footnote number in table footnote text.
tableFnoteRefPosition	integer	Placement of footnote reference in text: FnNumSuper: superscript, FnNumBase: baseline, FnNumSub: subscript.
tableFnoteRefPrefix	string	Prefix appearing before footnote reference in table cell.

Attribute	Type of value	Meaning
tableFnoteRefSuffix	string	Suffix appearing after footnote reference in table cell.
tableFnoteSuffix	string	Suffix appearing after footnote number in table footnote text.
textSelection	TxRange object	Currently selected text range or insertion point in the document. The insertion point is represented by a „range” of two equal locations.
tomboMarks	integer	True if registration marks are set to Tombo. When printing Tombo Marks, you must also set registrationMarks attribute (on page 24) to True.
topMargin	integer	Top page margin.
topMarginMin	integer	Minimum top margin allowed in the document.
trapwiseCompatible	integer	True if Trapwise Compatibility is enabled. Setting this to True automatically sets printToFile attribute (on page 24) to True and separations attribute (on page 24) to False.
useSepOverride	integer	True if separationOverride attribute (on page 79) overrides default.
viewBorders	integer	True if displaying of various otherwise invisible borders is enabled.
viewGrid	integer	True if grid lines are enabled.
viewMenu	FM object	Specific menu bar in viewer mode, if the document has one.
viewOnly	integer	True if the document is a view-only document.
viewRulers	integer	True if rulers are displayed.
viewSymbols	integer	True if otherwise invisible text symbols are displayed.
vSpreadLarge	integer	Vertical spread (in %) in large mathematic expressions.
vSpreadMedium	integer	Vertical spread (in %) in medium mathematic expressions.

Attribute	Type of value	Meaning
<code>vSpreadSmall</code>	integer	Vertical spread (in %) in small mathematic expressions.
<code>xRefs</code>	integer	Behavior of cross-references in the view only document. For list of possible values see "Cross-reference options in view only mode." on page 170.
<code>zoom</code>	integer	Zoom percentage of document (25% to 1600%).

4.2.25 Structural element

Table 36: Element has the following attributes.

Attribute	Type of value	Meaning
<code>attributeDisplay</code>	integer	Attribute display option for the element. For list of possible values see "Options for displaying element attributes in structural view." on page 140.
<code>attributes</code>	\N	Attributes of the element. Don't be confused by the name „attributes” - here it does mean the real attributes of the real structural element and not attributes of the wrapper Python object.
<code>bookComponent</code>	FM object	If the element represents a component file in an structured book, the associated Component.
<code>bookComponentMissing</code>	integer	True if a component file is missing from a book.
<code>collapsed</code>	integer	True if element is collapsed in Structure View.
<code>contextLabel</code>	string	The context label (if any) applied to the element.
<code>definition</code>	FM object	Definition of the element (ElementDef object).

Attribute	Type of value	Meaning
<code>errInBookComponent</code>	integer	True if there is a validation error for a component in a book.
<code>excluded</code>	integer	True if the element is excluded.
<code>firstChild</code>	FM object	Element's first child element (Element object).
<code>firstPgfClauses</code>	list of integers	First paragraph clauses (list of FormatClause objects) of the element's definition that apply to the element.
<code>formatOverride</code>	integer	True if the element has a format override.
<code>holeBefore</code>	integer	True if there are one or more elements missing before the element within its parent.
<code>invalidAttribute</code>	integer	True if the element contains an invalid attribute value.
<code>invalidHighestLevel</code>	integer	True if the element cannot be the highest-level element in the flow.
<code>invalidInParent</code>	integer	True if the element cannot occur anywhere in its current parent.
<code>invalidInPosition</code>	integer	True if the element is invalid in its current position.
<code>lastChild</code>	FM object	Element's last child element (Element object).
<code>lastPgfClauses</code>	list of integers	Last paragraph clauses (list of FormatClause objects) of the element's definition that apply to the element.
<code>looselyValid</code>	integer	True if the content is loosely valid (it can have some elements missing).
<code>markedDestination</code>	integer	True if a marked destination is generated for this element in PDF.
<code>mustBeEmpty</code>	integer	True if the element can't have any content.
<code>needContentAtBegin</code>	integer	True if content is needed at the beginning of the element.

Attribute	Type of value	Meaning
needContentAtEnd	integer	True if content is needed at end of the element. It is obsolete, but still supported for backward compatibility.
nextInvalid	FM object	Next invalid element in the document (Element object).
nextSibling	FM object	Next sibling of the element (Element object).
object	FM object	Object representing the element. The type of object depends on the element definition.
objectClauses	list of integers	Object clauses (list of FormatClause objects) of the element's definition that apply to the element.
parent	FM object	Element's parent element (Element object).
prefixClauses	list of integers	Prefix clauses (list of FormatClause objects) of the element's definition that apply to the element.
prevSibling	FM object	Element's previous sibling element (Element object).
specialCase	integer	True if the element is treated as a special case.
strictlyValid	integer	True if the content of the element is strictly valid.
suffixClauses	list of integers	Suffix clauses (list of FormatClause objects) of the element's definition that apply to the element.
textClauses	list of integers	Text clauses (list of FormatClause objects) of the element's definition that apply to the element.
textInvalidHere	integer	True if the element contains only text and the element definition disallows it. This attribute is obsolete.
textRange	TxRange object	Text range the element encompasses.

Attribute	Type of value	Meaning
type	integer	The type of element: ContainerElement, TableElement, MarkerElement, MathElement, XrefElement, TableTitleElement, TableHeaderElement, TableBodyElement, TableFooterElement, RowElement, CellElement, FootnoteElement, GraphicElement, VariableElement, RubiElement.
undefined	integer	True if the element is undefined.
unique	integer	Persistent unique identifier of the object.
userString	string	A string to which private data can be stored.
validationFlags	integer	Bit flags specifying the element's validity. To determine all the ways in which an element is invalid without querying all the validation properties, query this property. Each bit flag in the returned value represents the value of the validation property with the corresponding name: undefined attribute (on page 85), excluded attribute (on page 83), invalidInParent attribute (on page 83), invalidInPosition attribute (on page 83), mustBeEmpty attribute (on page 83), holeBefore attribute (on page 83), needContentAtBegin attribute (on page 83), needContentAtEnd attribute (on page 84), invalidHighestLevel attribute (on page 83), bookComponentMissing attribute (on page 82), errInBookComponent attribute (on page 83), invalidAttribute attribute (on page 83), strictlyValid attribute (on page 84), looselyValid attribute (on page 83), additional flags are AttributeValueRequired, InvalidText, TypeMismatch.

4.2.26 Definition of structural element

Table 37: ElementDef has the following attributes.

Attribute	Type of value	Meaning
<code>attributeDefs</code>	<code>\N</code>	The element definition's attribute definitions.
<code>autoInsert</code>	list of strings	List of tags of child elements that are automatically inserted when an element is initially added.
<code>comment</code>	string	Text string of comment.
<code>elementType</code>	integer	Type of formatter object representing the element with this definition. For list of possible values see "Types of structural elements." on page 147.
<code>errorOffsets</code>	list of integers	Contains offsets of errors in the general rule (two positions are specified only if the content rule is ambiguous).
<code>exclusions</code>	list of strings	List of elements that cannot occur within element with this definition.
<code>firstPgfrules</code>	list of integers	Format rules of the first paragraph (list of <code>FormatRule</code> objects).
<code>formatRules</code>	list of integers	Object format rules (list of <code>FormatRule</code> objects).
<code>generalRule</code>	string	Text of the element's general rule.
<code>inCatalog</code>	integer	True if the element definition is in the element catalog.
<code>inclusions</code>	list of strings	List of elements that can be included.
<code>lastPgfrules</code>	list of integers	Format rules of the last paragraph (list of <code>FormatRule</code> objects).
<code>name</code>	string	Name of the element definition.

Attribute	Type of value	Meaning
next	FM object	Next element definition in the document's list of element definitions (ElementDef object).
pgfFormat	string	The name of the paragraph format applied to the element.
prefixRules	list of integers	Prefix format rules (list of FormatRule objects).
structurePattern	string	The initial structure pattern; for table elements, a comma delimited string that specifies the necessary child elements to automatically insert.
suffixRules	list of integers	Suffix format rules (list of FormatRule objects).
tableTag	string	If the element type represents a table, the table format for new instances of the element.
textFormatRules	list of integers	Rules to format the text content (list of FormatRule objects).
validHighestLevel	integer	True if the element can be used as the highest-level element for a flow.

4.2.27 Ellipse

Table 38: Ellipse has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
smoothed	integer	True if smoothing is enabled. This property is always True for Ellipse objects. (Seems the ellipse is a specially smoothed rectangle only.)

4.2.28 Text flow

Table 39: F1ow has the following attributes.

Attribute	Type of value	Meaning
autoConnect	integer	True if autoconnect is enabled.
feather	integer	True if feather is enabled.
firstFrame	FM object	First text frame in flow (TxFrame object).
lastFrame	FM object	Last text frame in the flow (TxFrame object).
maxLineSpacing	integer	Maximum interline spacing.
maxPgfsSpacing	integer	Maximum interparagraph spacing.
name	string	Name of the flow tag.
next	FM object	Next flow in document (Flow object).
postScript	integer	True if flow's purpose is to contain PostScript code.
sideHead	integer	True if room is leaved for sideheads.
spacing	integer	Line spacing for synchronized baselines.
synchroMaxSize	integer	Maximum character height for synchronization of first line in column. If a character exceeds this height, FrameMaker product doesn't synchronize the first line.
synchronized	integer	True if baseline synchronization is enabled.
topElement	FM object	Highest-level element in flow (Element object).

4.2.29 Change of format

Table 40: FormatChange has the following attributes. See also: Table 5 on page 29, Table 9 on page 34.

Attribute	Type of value	Meaning
cellMarginChangeB	integer	Amount by which to increase or decrease the cell bottom margin.
cellMarginChangeL	integer	Amount by which to increase or decrease the cell left margin.
cellMarginChangeR	integer	Amount by which to increase or decrease the cell right margin.
cellMarginChangeT	integer	Amount by which to increase or decrease the cell top margin.
cellMarginFixedB	integer	True if the cell bottom margin is fixed.
cellMarginFixedL	integer	True if the cell left margin is fixed.
cellMarginFixedR	integer	True if the cell right margin is fixed.
cellMarginFixedT	integer	True if the cell top margin is fixed.
firstIndentChange	integer	Amount by which to increase or decrease the left margin of first line.
firstIndentIsRelative	integer	True if the indent of first line is relative to the left indent.
firstIndentRelativeLeft	integer	Position relative to left indent if firstIndentIsRelative attribute (on page 89) is True.
fontSizeChange	integer	Amount by which to increase or decrease the font size.
frameAboveIndented	integer	True if the position of the frame specified by frameAbove attribute (on page 35) is at the current left indent.
frameBelowIndented	integer	True if the position of the frame specified by frameBelow attribute (on page 35) is at the current left indent.

Attribute	Type of value	Meaning
<code>inCatalog</code>	integer	True if the format change list is in the catalog of format change lists. False if it is in an element definition, as part of the text format rules.
<code>leadingChange</code>	integer	Amount by which to increase or decrease the space below each line in paragraph.
<code>leftIndentChange</code>	integer	Amount by which to increase or decrease the left margin.
<code>lineSpacingFixed</code>	integer	True if the line spacing is fixed.
<code>moveTabs</code>	integer	Amount by which to move all tabulator positions in the paragraph.
<code>name</code>	string	Name of the format change list if it is in the catalog.
<code>next</code>	FM object	Next format change list in the document (FormatChange object).
<code>numTabs</code>	integer	The number of tabulators in the paragraph. To clear all the tabulators in the paragraph, set it to 0.
<code>paraFormat</code>	string	A paragraph format tag if the format change list specifies one. If this attribute is set, you can't change any of the other format change list attributes, except name attribute (on page 90).
<code>rightIndentChange</code>	integer	Amount by which to increase or decrease the right margin.
<code>spaceAboveChange</code>	integer	Amount by which to increase or decrease the space above the paragraph.
<code>spaceBelowChange</code>	integer	Amount by which to increase or decrease the space below.
<code>spreadChange</code>	integer	Amount by which to change the spread values.
<code>stretchChange</code>	integer	Amount by which to increase or decrease the character stretch expressed as a percentage.

4.2.30 Formatting rule

Table 41: `FormatRule` has the following attributes.

Attribute	Type of value	Meaning
<code>clause</code>	FM object	If the format rule is nested, the format rule clause that contains it (<code>FormatClause</code> object).
<code>clauses</code>	list of integers	The format rule's clause objects (list of <code>FormatClause</code> objects).
<code>elementDefinition</code>	FM object	If the format rule is not nested, the <code>ElementDef</code> object that contains it.
<code>elemsToCount</code>	list of strings	If the format rule is a level rule, the list of element tags to count among the element's ancestors.
<code>stopCounting</code>	string	If the format rule is a level rule, the tag of the element at which to stop counting elements.
<code>type</code>	integer	The format rule's type: <code>ContextRule</code> , <code>LevelRule</code> .

4.2.31 Formatting clause

Table 42: `FormatClause` has the following attributes.

Attribute	Type of value	Meaning
<code>changeList</code>	FM object	If the format rule clause specifies a format change list (<code>clause-Type</code> attribute (on page 92) specifies <code>ChangeListClause</code>), the associated format change list (<code>FormatChange</code> object). To change this property, use ???.

Attribute	Type of value	Meaning
<code>changeListTag</code>	string	If the format rule clause specifies a format change list (<code>clause-Type</code> attribute (on page 92) specifies <code>ChangeListTagClause</code>), the name of the format change list.
<code>clauseType</code>	integer	The type of rule clause: <code>TagClause</code> , <code>ChangeListTagClause</code> , <code>ChangeListClause</code> , <code>SubFormatClause</code> .
<code>contextLabel</code>	string	The context label for generated files. It cannot contain white-space characters or any of these special characters: () & , * + ? < > % [] = ! ; : { }
<code>formatTag</code>	string	The format tag if the format rule clause specifies one (<code>clause-Type</code> attribute (on page 92) specifies <code>TagClause</code>). If <code>isTextRange</code> attribute (on page 92) is <code>True</code> , it specifies a character format tag; otherwise it specifies a paragraph tag, table tag, marker type, cross-reference format, or equation size.
<code>isTextRange</code>	integer	<code>True</code> if the container element is formatted as a text range instead of a paragraph.
<code>prefixOrSuffix</code>	string	The text of the prefix or suffix or zero if there is no prefix or suffix.
<code>rule</code>	FM object	The format rule containing the format rule clause (<code>FormatRule</code> object).
<code>specification</code>	string	The format clause's context or level specification.
<code>subRule</code>	FM object	If the format rule clause contains a nested format rule (<code>clause-Type</code> attribute (on page 92) is set to <code>SubFormatClause</code>), the subrule (<code>FormatRule</code> object). To change this attribute, use ???

4.2.32 Footnote

Table 43: Footnote has the following attributes.

Attribute	Type of value	Meaning
<code>anchorLocation</code>	<code>TxLocation</code> object	Location of the footnote reference symbol.
<code>contentHeight</code>	integer	The distance between the top of the footnote and the baseline of the last line in the footnote.
<code>element</code>	FM object	If the footnote is in a structured flow, the <code>Element</code> object containing it.
<code>firstPgf</code>	FM object	First paragraph in the footnote (<code>Paragraph</code> object).
<code>inTextFrame</code>	FM object	Text frame containing the footnote (<code>TxFrame</code> object).
<code>inTextObject</code>	FM object	Text frame or table cell the footnote reference resides in (<code>TxFrame</code> or <code>TbCell</code> object).
<code>lastPgf</code>	FM object	Last paragraph in the footnote (<code>Paragraph</code> object).
<code>nextInDoc</code>	FM object	Next footnote (<code>Footnote</code> object) in the document.
<code>nextInFrame</code>	FM object	Next footnote in the text frame (<code>Footnote</code> object).
<code>number</code>	integer	Number of this footnote.
<code>overflowed</code>	integer	True if the text in the footnote overflows.
<code>prev</code>	FM object	Previous footnote in the text frame (<code>Footnote</code> object).
<code>unique</code>	integer	Persistent unique identifier of the object.

4.2.33 Group of objects

Table 44: Group has the following attributes.

Attribute	Type of value	Meaning
<code>firstGraphic</code>	FM object	First graphic object in the group.
<code>lastGraphic</code>	FM object	Last graphic object in the group.

4.2.34 Hidden page

Table 45: HiddenPage has the following attributes.

Attribute	Type of value	Meaning
<code>height</code>	integer	Height of page.
<code>name</code>	string	Name of the hidden page.
<code>pageFrame</code>	FM object	Page frame (UFrame object).
<code>width</code>	integer	Width of page.

4.2.35 Imported graphic

Table 46: GInset has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
<code>dpi</code>	integer	Scaling information for bitmap image file (corresponds to the value specified in the Image File Scaling Options dialog box when the graphics file is imported).
<code>editor</code>	string	Name of application to call to edit the inset or imported object.
<code>file</code>	string	Platform-specific pathname if the inset is an external inset, or a null string if it is internal. The path can be relative to document path.
<code>fixedSize</code>	integer	True if scaling of bitmap file is inhibited.
<code>flippedSideways</code>	integer	True if inset is flipped about the vertical axis.
<code>importHint</code>	string	Record identifying the filter used to import the graphic. The FrameMaker product uses this record to find the filter to use when updating the inset. For a more in-depth description of the syntax of this string, you should read the documentation available with FDK.
<code>updater</code>	string	This attribute is said to be unimplemented, but who knows?

4.2.36 Graphic line

Table 47: GLine has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
<code>numPoints</code>	integer	Number of vertices (= 2 in this case: the line's start and end point).

Attribute	Type of value	Meaning
<code>points</code>	list of coordinate (integer) pairs	List of (two) coordinate pairs that specify the line's vertices.

4.2.37 Marker

Table 48: Marker has the following attributes.

Attribute	Type of value	Meaning
<code>anchorLocation</code>	<code>TxLocation</code> object	Location of the marker symbol.
<code>element</code>	FM object	If the marker is structured, the <code>Element</code> object containing it.
<code>next</code>	FM object	Next marker in the document (<code>Marker</code> object).
<code>text</code>	string	The marker's text content.
<code>type</code>	FM object	Type of the marker (<code>MarkerType</code> object).
<code>unique</code>	integer	Persistent unique identifier of the object.

4.2.38 Type of marker

Table 49: MarkerType has the following attributes.

Attribute	Type of value	Meaning
invariantName	string	An internal name for the marker type. By default, this is the same as name attribute (on page 97). However, this can differ from it if the user interface is in another language.
name	string	Name of the marker type, as it appears in the user interface.
next	FM object	Next marker type (MarkerType object).
oldTypeNum	integer	In documents earlier than version 5.5 there were type numbers instead of marker types. When opening such older documents markers with type numbers matching this attribute will be converted to markers of this type.
public	integer	True if the marker type should appear in the user interface.
required	integer	True if the marker type is required by FrameMaker products.
transient	integer	True if markers of this type should not be saved to files.

4.2.39 Master page

Table 50: MasterPage has the following attributes.

Attribute	Type of value	Meaning
height	integer	Height of page.
name	string	Name of the master page (for example, Right or Left).

Attribute	Type of value	Meaning
next	FM object	Next master page (MasterPage object) in the document.
number	integer	Page number.
pageFrame	FM object	Page frame (UFrame object).
prev	FM object	Previous master page (MasterPage object) in the document.
width	integer	Width of page.

4.2.40 Mathematic expression

Table 51: Expression has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
alignment	integer	Type of text line: TextlineLeft, TextlineRight, TextlineCenter, TextlineExpression.
originX	integer	Horizontal placement of expression base point relative to left side of text frame.
originY	integer	Vertical placement of expression base point relative to top of text frame.
size	integer	Equation size: LargeExpression, MediumExpression, SmallExpression.
source	string	String representing the expression. The syntax is described in the handbook on MIF format.

4.2.41 Menu

Table 52: Menu has the following attributes. See also: Table 8 on page 33.

Attribute	Type of value	Meaning
<code>firstItem</code>	FM object	The first menu item or menu in the menu.
<code>type</code>	integer	Type of menu. For list of possible values see "Types of menu." on page 159.

4.2.42 Paragraph of text

Table 53: Paragraph has the following attributes. See also: Table 5 on page 29, Table 9 on page 34.

Attribute	Type of value	Meaning
<code>fixedMargins</code>	integer	Specifies which cell margins are fixed. The following values can be bitwise ORed (Python operator <code>&</code>) into it: <code>FixedBottomMargin</code> : the bottom margin is fixed, <code>FixedLeftMargin</code> : the left margin is fixed, <code>FixedRightMargin</code> : the right margin is fixed, <code>FixedTopMargin</code> : the top margin is fixed. If the margin for a cell is fixed, the margin property specifies the absolute value of the cell margin. For example, if <code>FixedBottomMargin</code> is set, <code>cellMarginBottom</code> attribute (on page 35) specifies the absolute value of the cell's bottom margin, overriding the cell margin specified by the table format. If <code>FixedBottomMargin</code> is not set, <code>cellMarginBottom</code> attribute (on page 35) is added to the margin specified by the table format.
<code>fontEncoding</code>	string	Encoding of the font.

Attribute	Type of value	Meaning
fontPlatformName	string	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
fontPSName	string	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
formatOverride	integer	True if format of the paragraph differs from corresponding paragraph format.
inTextFrame	FM object	Text frame containing first line of the paragraph (TxFrame object).
inTextObject	FM object	Subcolumn, footnote or table cell the paragraph begins in (TxColumn, Footnote or TbCell object).
lineSpacing	integer	Type of spacing between lines in a paragraph. For list of possible values see "Options how to compute height of lines in text." on page 158.
locked	integer	True if the paragraph is part of a text imported by reference that retains formatting information from the source document. The paragraph is not affected by global formatting performed on the document.
name	string	Name of paragraph format used.
nextInDoc	FM object	Next paragraph in the document (Paragraph object).
nextInFlow	FM object	Next paragraph in the flow (Paragraph object).
nextTag	string	Tag for new next paragraph.
number	string	The formatted string representation of the paragraph number; for example, 1.2 for a paragraph whose autoNumFormat attribute (on page 35) is set to <n>.<n+>.
numTabs	integer	Number of tabulators in the paragraph.

Attribute	Type of value	Meaning
prevInFlow	FM object	Previous paragraph in the flow (Paragraph object).
spellChecked	integer	True if paragraph has been spell-checked.
unique	integer	Persistent unique identifier of the object.
useNextTag	integer	True if next paragraph tag should be used.

4.2.43 Paragraph format

Table 54: ParaFormat has the following attributes. See also: Table 5 on page 29, Table 9 on page 34.

Attribute	Type of value	Meaning
bookmarkLevel	integer	The outline level of paragraphs with the format when the Frame-Maker product generates data for PDF. If it is set to 0, the Frame-Maker product does not generate bookmarks for paragraphs with this format. If it is greater than 0, the FrameMaker product sets the outline level of paragraphs with this format to the specified level.

Attribute	Type of value	Meaning
<code>fixedMargins</code>	integer	Specifies which cell margins are fixed. The following values can be bitwise ORed (Python operator <code>&</code>) into it: <code>FixedBottomMargin</code> : the bottom margin is fixed, <code>FixedLeftMargin</code> : the left margin is fixed, <code>FixedRightMargin</code> : the right margin is fixed, <code>FixedTopMargin</code> : the top margin is fixed. If the margin for a cell is fixed, the margin property specifies the absolute value of the cell margin. For example, if <code>FixedBottomMargin</code> is set, <code>cellMarginBottom</code> attribute (on page 35) specifies the absolute value of the cell's bottom margin, overriding the cell margin specified by the table format. If <code>FixedBottomMargin</code> is not set, <code>cellMarginBottom</code> attribute (on page 35) is added to the margin specified by the table format.
<code>fontEncoding</code>	string	Encoding of the font.
<code>fontPlatformName</code>	string	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
<code>fontPSName</code>	string	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
<code>lineSpacing</code>	integer	Type of spacing between lines in a paragraph. For list of possible values see "Options how to compute height of lines in text." on page 158.
<code>markedDestination</code>	integer	True if a marked destination is generated for paragraphs having this format in PDF.
<code>name</code>	string	Name of the paragraph format.
<code>next</code>	FM object	Next paragraph format in the document (<code>ParaFormat</code> object).
<code>nextTag</code>	string	Tag for new next paragraph.

Attribute	Type of value	Meaning
numTabs	integer	Number of tabulators in the paragraph.
structureLevel	integer	Structure level of paragraphs having this format in PDF. This value is used when the pdfStructure attribute (on page 77) of the document is True and the FrameMaker product generates PDF data.
useNextTag	integer	True if next paragraph tag should be used.

4.2.44 Polygon

Table 55: Polygon has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
numPoints	integer	Number of polygon's vertices.
points	list of coordinate (integer) pairs	List of coordinate pairs that specify polygon's vertices.
smoothed	integer	True if polygon is smoothed.

4.2.45 Polyline

Table 56: Polyline has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
numPoints	integer	Number of polyline's vertices.

Attribute	Type of value	Meaning
points	list of coordinate (integer) pairs	List of coordinate pairs that specify polyline's vertices.
smoothed	integer	True if polyline is a Bezier curve.

4.2.46 Rectangle

Table 57: Rectangle has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
smoothed	integer	True if smoothing is enabled.

4.2.47 Reference page

Table 58: RefPage has the following attributes.

Attribute	Type of value	Meaning
height	integer	Height of page.
name	string	Name of the reference page.
next	FM object	Next reference page (RefPage object) in the document.
number	integer	Page number.
pageFrame	FM object	Page frame (UFrame object).
prev	FM object	Previous reference page (RefPage object) in the document.

Attribute	Type of value	Meaning
width	integer	Width of page.

4.2.48 Rectangle with rounded corners

Table 59: RoundRect has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
radius	integer	Radius of corner; 0 for a square corner.

4.2.49 Table row

Table 60: TbRow has the following attributes.

Attribute	Type of value	Meaning
conditions	list of integers	Conditions applied to the row (list of CondFormat objects).
conditionShown	integer	True if the row's condition is shown.
element	FM object	If the row is in a structured document, the Element object containing the row.
firstCell	FM object	First cell in row (TbCell object).
height	integer	Height of the row.
keepWithNext	integer	True if the row should be kept with next one.
keepWithPrev	integer	True if the row should be kept with previous one.

Attribute	Type of value	Meaning
locX	integer	Offset from the left side of the text frame containing the row.
locY	integer	Offset from the top of the page frame containing the row.
maxHeight	integer	Maximum row height (independent of content).
minHeight	integer	Minimum row height (independent of content).
next	FM object	Next row (TbRow object) in the table.
prev	FM object	Previous row (TbRow object) in the table.
separationOverride	FM object	Color separation format override (Color object).
shown	integer	True if the conditional row is shown.
start	integer	Row placement: PlaceRowAnywhere, RowAtTopOfColumn, RowAtTopOfPage, RowAtTopOfLeftPage, RowAtTopOfRightPage.
styleOverride	integer	Bit flags that specify which overrides (condition indicators) are used: RowCSDoubleUnderline, RowCSNoOverride, RowCSSOverline, RowCSSingleUnderline, RowCSSStrikethrough, RowCSChangeBar, RowCSNumericUnderline. All style condition indicators are represented as hatched lines for the table rows.
table	FM object	Table containing the row (Table object).
type	integer	Type of row: HeadingRow, BodyRow, FootingRow.
useSepOverride	integer	True if separationOverride attribute (on page 106) overrides default from the table.
width	integer	Width of the row.

4.2.50 Rubi

Table 61: Rubi has the following attributes.

Attribute	Type of value	Meaning
element	FM object	If the rubi group is in a structured document, the object handle of the associated Element for the rubi element.
groupElement	FM object	If the rubi group is in a structured document, the Element object for the rubi group element.
next	TxRange object	Next instance of a rubi composite (Rubi object) in the document.
oyamojiRange	TxRange object	The text range the oyamoji text encompasses.
range	TxRange object	The text range the rubi text encompasses.
unique	integer	Persistent unique identifier of the object.

4.2.51 Format of ruling

Table 62: RulingFormat has the following attributes.

Attribute	Type of value	Meaning
color	FM object	Spot color of ruling format (Color object).
gap	integer	Gap between double ruling lines (0.015 pt to 360 pt).
lines	integer	Number of ruling lines (0 to 2 lines).
name	string	Name of the ruling format.

Attribute	Type of value	Meaning
next	FM object	Next ruling format in the document (RulingFormat object).
pattern	integer	The fill pattern (numbers between 0 and 15). For list of possible values see "Fill patterns" on page 138.
penWidth	integer	Ruling line thickness (0.015 pt to 360 pt).

4.2.52 Frame product session

Table 63: Session has the following attributes. See also: Table 2 on page 25, Table 3 on page 26.

Attribute	Type of value	Meaning
accessBar	integer	True if the QuickAccess bar is visible.
activeBook	FM object	The book with input focus (Book object).
activeDoc	FM object	The document with input focus (Document object).
addMarkerType	string	The name of a marker type to add to the list of marker type names. This property is write-only; its value is an empty string.
autoBackup	integer	True if automatic backup is enabled.
autoSave	integer	True if automatic save is enabled.
autoSavePeriod	integer	Time between automatic saves in seconds (60 seconds to 10800 seconds).
binDir	string	Pathname of \$FMHOME/bin directory on UNIX, the same on Windows even when the directory does not exist, or the Modules folder on the Macintosh.

Attribute	Type of value	Meaning
currentDir	string	Pathname of the directory from which the FrameMaker product was started (on UNIX and Windows) or the Modules folder (on the Macintosh).
currentMenuSet	integer	Type of menu set: QuickMenu, CompleteMenu, CustomMenu.
displaying	integer	False if screen refresh is completely turned off. Note in this mode many measures returned may be false.
displayName	string	The name of the display on which the FrameMaker product session is running (UNIX only).
docIconBar	integer	True if the buttons that appear on the right side of the document or book window above the vertical scroll bar are on. Changing this property affects only books and documents that are opened subsequently; it does not affect those already open.
elementReformatting	integer	True if element reformatting is enabled (FrameMaker+SGML only).
exportFilters	list of strings	List of hint strings of export filters available in the current session.
firstCommand	FM object	First command in the list of commands in the session (Command object).
firstMenuItem	FM object	First menu item or menu in the list of menus, menu items, and menu item separators in the session (Command, Menu or Separator object).
firstOpenBook	FM object	First open book (Book object) in session.
firstOpenDoc	FM object	First open document (Document object) in session.
fontAngleNames	list of strings	List of font angles available in the current session.

Attribute	Type of value	Meaning
fontFamilyAttribs	list of integers	List of flags that indicate properties for each font family listed by fontFamilyNames attribute (on page 110). This list of integers is indexed the same as the list of font family names, and corresponds directly to that list. Each number is a packed field; the high order 16 bits indicate a surrogate font, and the low order bits indicate attributes for the font family. The flags and their meaning follow: FamilyVisible: Family is visible in menu. FamilySelectable: Family can be selected in menu. FamilyMapped: Family is always mapped to another family. SurrogateFamily: The family mapped to if FamilyMapped bit is set. To get index of the family you can bitwise AND the value with SurrogateFamily and then shift it right by 16 bits (,... & maker.symbol.SurrogateFamily >> 16”).
fontFamilyNames	list of strings	List of font family names available in the current session. Note that this list does not include combined fonts (CombiFont).
fontVariations	list of strings	List of font variations available in the current session.
fontWeights	list of strings	List of font weights available in the current session.
formattingBar	integer	True if the formatting bar is visible (Windows only).
gravity	integer	True if gravity is turned on for the session.
greekSize	integer	Size at which to greek text.
helpDir	string	Pathname of the FrameMaker product help directory.
homeDir	string	Pathname of \$FMHOME directory (on UNIX or Windows) or the FrameMaker product folder (on the Macintosh).
hostName	string	Name of the host computer.
importFilters	list of strings	List of hint strings of import filters available in the current session.

Attribute	Type of value	Meaning
initDir	string	Pathname of \$FMHOME/fminit directory (on UNIX or Windows) or the Modules folder (on the Macintosh).
language	integer	Product language, one of: LangBritish, LangEnglish, LangFrench, LangGerman, LangItalian, LangNone, LangSpanish, LangSwedish, LangJapanese, LangChineseTraditional, LangChineseSimplified, LangKorean
majorVersion	integer	Frame product major version number (before the decimal).
markerNames	list of strings	List of standard marker types for the current session. For versions prior to 5.5, this returned the list of all marker types for the current session. In version 5.5, marker types were assigned to the document; use the markerNames attribute (on page 73) of Document to get the full list of marker types.
minorVersion	integer	Frame product minor version number (after the decimal).
openDir	string	Directory pathname of last opened document or book (on UNIX) or the FrameMaker product directory (on the Macintosh and Windows).
path	string	Pathname to search to start the FrameMaker product (on UNIX or Windows) or the FrameMaker product folder (on the Macintosh).
platform	string	Name of the platform on which the current session is running: Sun, HP, RS6000, Alpha, SGI, Macintosh, Intel.
processNum	integer	The process number of the FrameMaker product session (UNIX only).

Attribute	Type of value	Meaning
product	string	FrameMaker product name: FrameMaker, Frame-Maker+SGML, FrameViewer, DemoMaker, Demo-Maker+SGML.
reformatting	integer	True if reformatting is enabled.
rememberFonts	integer	True if remembering of missing font names is activated .
rpcName	string	The RPC name of the FrameMaker product session (UNIX only).
rpcNumber	integer	The RPC process number of the FrameMaker product session (UNIX only).
snap	integer	True if snap is turned on for the session.
system	string	Operating system under which the current session is running: Solaris, SunOS, HPUX, AIX, OSF1, IRIX, Macintosh, and DOS (instead of Windows).
tmpDir	string	Pathname of temporary directory for internal FrameMaker product processes; the FrameMaker product directory (on the Macintosh); the directory specified by the DOS TEMP environment variable (on Windows); the directory specified by \$TMPDIR (on UNIX).
userHomeDir	string	Pathname of the user's home directory (UNIX) or the product directory.
userLogin	string	User login name.
userName	string	User name.
validation	integer	True if validation of structure is enabled (FrameMaker+SGML only).
windowSystem	string	Name of window system that the FrameMaker product is running under: Macintosh, MSWindows, X Windows.

4.2.53 Column of text

Table 64: TxColumn has the following attributes.

Attribute	Type of value	Meaning
contentHeight	integer	The distance between the top of the column and the baseline of the last line in the column.
firstAFrame	FM object	First frame anchored in the column (AFrame object).
firstCell	FM object	First table cell in the column (TbCell object).
firstFootnote	FM object	First footnote in the column (Footnote object).
firstPgf	FM object	First paragraph in the column (Paragraph object).
frameParent	FM object	Parent of the text frame containing the column (AFrame or UFrame object).
height	integer	Height of the column.
lastAFrame	FM object	Last anchored frame in the column (AFrame object).
lastCell	FM object	Last table cell in the column (TbCell object).
lastFootnote	FM object	Last footnote in the column (Footnote object).
lastPgf	FM object	Last paragraph in the column (Paragraph object).
locX	integer	Offset from left side of the text frame that contains the column.
locY	integer	Offset from top of text frame that contains the column.
next	FM object	Next column in the same text frame (TxColumn object).
overflowed	integer	True if the text frame containing the column has Autoconnect turned off and text overflows the column.

Attribute	Type of value	Meaning
parent	FM object	Text frame containing the column (TxFrame object).
prev	FM object	Previous column within the same text frame (TxColumn object).
unique	integer	Persistent unique identifier of the object.
width	integer	Width of the column.

4.2.54 Table

Table 65: Table has the following attributes. See also: **Table 11** on page 39.

Attribute	Type of value	Meaning
anchorLocation	TxLocation object	Location of the table anchor.
bodyElement	FM object	Element containing body rows of the table.
colWidths	list of integers	List of column widths.
container	FM object	The element containing the table.
contentHeight	integer	The height of the table title.
element	FM object	If the table is in a structured document, the Element object associated with the table.
firstPgf	FM object	If table has a title, the first paragraph in the title (Paragraph object).
firstRow	FM object	First row in the table (TbRow object).
footerElement	FM object	Structural element containing footer rows of the table.

Attribute	Type of value	Meaning
headerElement	FM object	Structural element containing header rows of the table.
lastPgf	FM object	If table has a title, the last paragraph in the title (Paragraph object).
lastRow	FM object	Last row in the table (TbRow object).
locked	integer	True if the table is part of a text inset that retains formatting information from the source document. The table is not affected by global formatting performed on the document.
next	FM object	Next table (Table object) in the document.
numCols	integer	Number of columns in the table.
numRows	integer	Number of rows in the table.
overflowed	integer	True if the table has cells that are not shown because they extend beyond the text frame boundaries.
selectedColLeft	integer	If (a portion of) table is selected, number of leftmost selected column (columns are numbered from left to right, starting with 0).
selectedColRight	integer	Number of rightmost selected column, if the table is selected (columns are numbered from left to right, starting with 0).
selectedRowBottom	FM object	Bottom row in selection, if table is selected (TbRow object).
selectedRowTop	FM object	Top row in selection, if (a portion of) table is selected (TbRow object).
titleElement	FM object	Structural element containing the table title.
titleSelected	integer	True if the table title is selected.
unique	integer	Persistent unique identifier of the object.

Attribute	Type of value	Meaning
width	integer	Width of the table.

4.2.55 Format of table

Table 66: TbFormat has the following attributes. See also: Table 11 on page 39.

Attribute	Type of value	Meaning
bodyRows	integer	Number of body rows for new table.
columns	integer	Number of columns for new table.
footerRows	integer	Number of footing rows for new table.
headerRows	integer	Number of heading rows for new table.
next	FM object	Next table format in the document (TbFormat object).
titleName	string	Name of the paragraph format of the table title.

4.2.56 Text frame

Table 67: TxFrame has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
balanceColumns	integer	True if terminal and underfilled columns in the flow are balanced.
columnGap	integer	Size of gap between columns (0 to 50 inches).

Attribute	Type of value	Meaning
firstAFrame	FM object	First frame anchored in the text frame (AFrame object).
firstCell	FM object	First table cell in the text frame (TbCell object).
firstColumn	FM object	First column in the text frame (TxColumn object).
firstFootnote	FM object	First footnote in the text frame (Footnote object).
firstPgf	FM object	First paragraph in the text frame (Paragraph object).
flow	FM object	Flow containing the text frame (Flow object).
isButton	integer	True if the text frame is a hypertext button.
lastAFrame	FM object	Last anchored frame in the text frame (AFrame object).
lastCell	FM object	Last table cell in the text frame (TbCell object).
lastColumn	FM object	Last column in the text frame (TxColumn object).
lastFootnote	FM object	Last footnote in the text frame (Footnote object).
lastPgf	FM object	Last paragraph in the text frame (Paragraph object).
next	FM object	Next text frame in the flow (TxFrame object).
numCols	integer	The number of columns in the text frame (1-10).
prev	FM object	Previous text frame in the flow (TxFrame object).
sideHeadGap	integer	Gap between side head area and body text area (0 to 50 inches).
sideHeadPlace	integer	Placement of side heads relative to columns in the text frame: SideHeadLeft, SideHeadRight, SideHeadIn, SideHeadOut.
sideHeadWidth	integer	Width of side head area for the text frame (0 to 50 inches).

4.2.57 Graphic text object

Table 68: TxLine has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
alignment	integer	Alignment of text line: TextlineLeft, TextlineRight, TextlineCenter, TextlineExpression.
originX	integer	Horizontal placement of text line base point relative to left side of frame.
originY	integer	Vertical placement of text line base point relative to top of frame.

4.2.58 Text imported by an extension module

Table 69: PlugInset has the following attributes. See also: Table 10 on page 37.

Attribute	Type of value	Meaning
clientData	string	Data used by the plug-in (for example, a SQL query).
clientName	string	Registered name of the plug-in that created the inset.
source	string	The name that appears as the source in the text inset properties dialog box.
type	string	The name that appears as the source type in the text inset properties dialog box.
unresolved	integer	True if the inset is unresolved. A plug-in should set this property to True if it is unable to resolve the inset.

4.2.59 Text imported from a flow

Table 70: FInset has the following attributes. See also: Table 10 on page 37.

Attribute	Type of value	Meaning
flowName	string	Name of the imported flow if mainFlow attribute (on page 119) is False. If the source file is an edition, it is set to Macintosh edition. ???
formatting	integer	Formatting option of the imported text. For list of possible values see "Options for formatting texts imported by reference." on page 168.
mainFlow	integer	True if the inset text is imported from the main flow of the source document.
removeOverrides	integer	True if format overrides are removed from the text when formatting attribute (on page 119) is set to TargetFormat.
removePageBreaks	integer	True if page breaks are removed from the text when formatting attribute (on page 119) is set to TargetFormat.
sourcePages	integer	Type of pages the imported flow is on: BodyPageFlow, ReferencePageFlow.

4.2.60 Text imported by reference

Table 71: TxInset has the following attributes. See also: Table 10 on page 37.

Attribute	Type of value	Meaning
encoding	string	???

Attribute	Type of value	Meaning
<code>lineIsParagraph</code>	integer	True if line ends in the imported text file are treated as paragraph ends.

4.2.61 Table imported by reference

Table 72: `TbInset` has the following attributes. See also: Table 10 on page 37.

Attribute	Type of value	Meaning
<code>byRows</code>	integer	True if each paragraph of the imported text is converted to a row of table cells; False if each paragraph of the imported text is converted to a table cell.
<code>emptyHeaders</code>	integer	True if the imported text is not used to fill the heading rows.
<code>encoding</code>	string	???
<code>name</code>	string	The table format tag of the imported table.
<code>numCols</code>	integer	If <code>byRows</code> attribute (on page 120) is False, the number of columns in the table.
<code>numHeaderRows</code>	integer	The number of heading rows in the table.
<code>numSeparators</code>	integer	If <code>separator</code> attribute (on page 120) specifies a space, the number of spaces used as a separator to parse the text into table cells.
<code>separator</code>	string	If <code>byRows</code> attribute (on page 120) is True, a string specifying a separator, such as a tab, used to parse the text into table cells.

4.2.62 Unanchored frame

Table 73: UFrame has the following attributes. See also: Table 6 on page 30, Table 7 on page 33.

Attribute	Type of value	Meaning
<code>firstGraphic</code>	FM object	First object in the frame (backmost object).
<code>lastGraphic</code>	FM object	Last object in the frame (frontmost object).
<code>name</code>	string	If a reference frame, the frame's name.
<code>page</code>	FM object	If the frame is a page frame, the page that it belongs to (HiddenPage, BodyPage, MasterPage, or RefPage object).

4.2.63 Variable

Table 74: Variable has the following attributes.

Attribute	Type of value	Meaning
<code>element</code>	FM object	If the variable is in a structured document, the Element object associated with the variable.
<code>format</code>	FM object	The variable instance's format (VarFormat object).
<code>locked</code>	integer	True if the variable is included in a text inset that gets its formatting from the source document. The variable is not affected by global formatting performed on the document.
<code>next</code>	FM object	Next variable instance (Variable object) in the document.
<code>textRange</code>	TxRange object	Text range the variable instance encompasses.

Attribute	Type of value	Meaning
unique	integer	Persistent unique identifier of the object.

4.2.64 Variable format

Table 75: VarFormat has the following attributes.

Attribute	Type of value	Meaning
definition	string	The variable format definition (a string specifying building blocks and text).
name	string	Name of the variable format.
next	FM object	Next variable format (VarFormat object) in the document.
type	integer	Type of the variable format. If set to UserVariable: a user-defined variable format. The following types specify system formats: CurrentPageNum, PageCount, CurrentDateLong, CurrentDateShort, ModificationDateLong, ModificationDateShort, CreationDateLong, CreationDateShort, FileNameLong, FileNameShort, HeaderFooter1, HeaderFooter2, HeaderFooter3, HeaderFooter4, TableContinuation, TableSheet.

4.2.65 Cross-reference

Table 76: XRef has the following attributes.

Attribute	Type of value	Meaning
element	FM object	If the cross-reference is in a structured document, the Element object associated with the element.
file	string	Name of the file containing the cross-reference source. If the cross-reference source is in the same document as the cross reference, the filename is an empty string.
format	FM object	Format of the cross-reference (XRefFormat object).
locked	integer	True if the cross-reference is part of a text inset that retains formatting information from the source document. The cross-reference is not affected by global formatting performed on the document.
next	FM object	Next cross-reference instance in document (XRef object).
sourceIsElement	integer	True if the cross-reference source is a structural element.
sourceText	string	If sourceIsElement attribute (on page 123) is False, the text of the cross-reference source marker; if sourceIsElement attribute (on page 123) is True, a string in the form <code>UID:src-name:text</code> , where UID is the unique attribute (on page 85), name is the element name, and text is additional text of the element that is the cross-reference source.
textRange	TxRange object	Text range the cross-reference instance encompasses.
unique	integer	Persistent unique identifier of the object.

Attribute	Type of value	Meaning
unresolved	integer	True if the FrameMaker product was unable to resolve the cross-reference the last time it updated cross-references. Note that this property is set only when the FrameMaker product updates cross-references. Changes to the document, in and of themselves, do not affect this property.

4.2.66 Cross-reference format

Table 77: XRefFormat has the following attributes.

Attribute	Type of value	Meaning
definition	string	The cross-reference format (a string specifying building blocks and text).
name	string	Name of the cross-reference format.
next	FM object	Next cross-reference format in the document (XRefFormat object).

4.2.67 Text

Table 78: TxLocation has the following attributes. See also: Table 5 on page 29.

Attribute	Type of value	Meaning
baseLine	integer	The vertical offset of the line containing the location's baseline, measured from the top of the object containing the text.

Attribute	Type of value	Meaning
charTag	string	Name of character format applied to the location.
conditions	list of integers	Conditions applied to the text at the location (list of CondFormat objects).
fontPlatformName	string	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
fontPSName	string	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
height	integer	Height of text at text location.
inTextFrame	FM object	Text frame containing the text (TxFrame object).
inTextObject	FM object	Text frame or text line the text appears in (TxFrame or TxLine object).
lineAscent	integer	The ascent of the line containing the location, measured from the line's baseline.
lineDescent	integer	The descent of the line containing the location, measured from the line's baseline.
locked	integer	True if the text at the location is included in a text inset that gets its formatting from the source document.
locX	integer	Offset of the location from the left side of the column (TxColumn object) containing it.
locY	integer	Offset of the top of a character to the right of the location from the top of the subcolumn (TxColumn object) containing it.
separationOverride	FM object	Color for separation override at the text location (Color object).

Attribute	Type of value	Meaning
styleOverride	integer	Bit flags that specify which overrides (condition indicators) are used: RowCSDoubleUnderline, RowCSNoOverride, RowCSOverline, RowCSSingleUnderline, RowCSStrikethrough, RowCSChangeBar, RowCSNumericUnderline.
useSepOverride	integer	True if separationOverride attribute (on page 125) overrides default at the location.
width	integer	Width of a character on the right-hand side of the location.

Chapter 5

Procedural objects

This chapter describes procedural objects related to the module `maker`.

5.1 Functions

5.1.1 Functions of the module `maker`.

5.1.1.1 `TxLocation()`

Description. Creates new `TxLocation` object.

Arguments. First argument is a `Paragraph` or `TxLine` object. Such objects can be found or created at various places within the structure of a `Document` object. Second argument is an offset within the first argument object. Offset of zero points to the left of the first character in the object, offset of 1 to the left of the second character and so on. Offset of `EndOffset` points to the right of the last character, which is the paragraph sign. Offset of `EndOffset - 1` points to

the left of the last character and so on.

Return value. New location in text (`TxLocation` object).

Exceptions. `TypeError` or `ValueError` is raised when there is something wrong with parameters.

5.1.1.2 `TxRange()`

Description. Creates new `TxRange` object.

Arguments. Two locations in text (`TxLocation` objects).

Return value. The newly created text range (`TxRange` object). Validity of the result is not guaranteed - the range is useless for example if arguments point to different text flows.

Exceptions. `TypeError` or `ValueError` when there is something wrong with supplied parameters.

5.2 Methods

5.2.1 Methods of wrapper objects

5.2.1.1 `addText()`

Description. Adds text string in `Frame`'s encoding at the location in text.

Arguments. String in `FrameMaker` character set (described in one of online manuals).

Return value. New location in text pointing just after the last added character.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `TxLocation`.

5.2.1.2 applyLayout()

Description. Applies layout of a page to the receiver.

Arguments. One of the page objects (BodyPage, MasterPage or RefPage).

Return value. Nothing.

Exceptions. ValueError is generated when an argument of different type was provided.
RuntimeError is generated when an error occurs in FrameMaker.

Binding. Bound to: BodyPage, MasterPage, RefPage.

5.2.1.3 close()

Description. Closes a document, book or session. If the object to be closed was modified and the parameter is not 1, nothing happens.

Arguments. One integer - the value of zero causes abort of the operation when the object to be closed is unsaved, the value of one permits to close even unsaved objects.

Return value. Nothing.

Exceptions. RuntimeError is generated when closing unsaved modified object and the argument was zero.

Binding. Bound to: Book, Document, Session.

5.2.1.4 copy()

Description. Copies the current selection to the clipboard.

Arguments. Ignored.

Return value. Nothing.

Exceptions. RuntimeError is generated when an error occurs in FrameMaker.

Binding. Bound to: Document.

5.2.1.5 cut()

Description. Cuts the current selection to the clipboard.

Arguments. Ignored.

Return value. Nothing.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: Document.

5.2.1.6 delete()

Description. This is a way how to remove objects from documents and books. The receiver and all objects it may contain will be removed. It is wise not to use deleted objects any more. Some objects can't be deleted, as Left and Right master pages, page frames, predefined colors and formats of system variables.

Arguments. Ignored.

Return value. Nothing.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `AFrame`, `Arc`, `BodyPage`, `Component`, `CharFormat`, `Color`, `Comb-iFont`, `Command`, `CondFormat`, `Element`, `ElementDef`, `Ellipse`, `Flow`, `FormatChange`, `FormatRule`, `FormatClause`, `Footnote`, `Group`, `GInset`, `GLine`, `Marker`, `MarkerType`, `MasterPage`, `Expression`, `Menu`, `Separator`, `Paragraph`, `ParaFormat`, `Polygon`, `Polyline`, `Rectangle`, `RefPage`, `RoundRect`, `TbRow`, `Rubi`, `RulingFormat`, `TxColumn`, `Table`, `TbFormat`, `TxFrame`, `TxLine`, `PlugInset`, `FInset`, `TxInset`, `TbInset`, `UFrame`, `Variable`, `VarFormat`, `XRef`, `XRefFormat`, `TxRange`.

5.2.1.7 display()

Description. Accommodates the displayed area of the range's document so that the nearest end of it

is visible in the window. Using this method within inactive documents still does not guarantee visibility - you may want to set the `activeDoc` attribute (on page 108).

Arguments. Ignored.

Return value. Nothing.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `TxRange`.

5.2.1.8 `execF()`

Description. Every function the `FrameMaker` engine performs has a code assigned. This method allows to execute series of functions by codes.

Arguments. List of function codes (integers).

Return value. Nothing.

Exceptions. `TypeError` is generated when the argument is not a list. `ValueError` is generated when a member of the list is not integer. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `Session`.

5.2.1.9 `foundIn()`

Description. This method allows a menu related object (`Command`, `Menu` or `Separator`) to search for itself in a menu.

Arguments. Menu object to search in and an optional integer (defaulting to zero). When this second parameter is non-zero, the search is recursive.

Return value. The Menu object the receiver is directly in.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `ValueError` is generated when the first argument is not a Menu object. `RuntimeError` is generated when an error occurs in `FrameMaker` or the menu item was not found within the argument.

Binding. Bound to: Command, Menu, Separator.

5.2.1.10 `getNamed()`

Description. Searches for objects having a name. Specifically all formats and all objects related to menu have names, and in addition to them: Color, CombiFont, ElementDef, FormatChange, MasterPage, RefPage and UFrame (on a reference page).

Arguments. One integer - code of object's subtype (For list of possible values see "Codes of object classes." on page 142.) and one string - the name.

Return value. The object found.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: Document, Session.

5.2.1.11 `Import()`

Description. Imports text or graphics into a document. When importing by reference, objects of the following types are created: `GInset`, `TxInset`, `TbInset`, `FInset`.

Arguments. Pathname of the file to import and dictionary of additional options that will influence the importing process.

Return value. The newly created object.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`. Additional information is written to the variable `maker.ioStatus` in the form of list of strings.

Binding. Bound to: `TxLocation`.

5.2.1.12 `newAnchor()`

Description. Creates new anchored object with anchor at the location in text. It can create objects of the following types: `AFrame`, `Footnote`, `Marker`, `PlugInset` and `Table`.

Arguments. One integer - code of object's subtype. For list of possible values see "Codes of object classes." on page 142. (Not all codes are applicable.)

Return value. The newly created object.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `TxLocation`.

5.2.1.13 `newGraphic()`

Description. Creates new graphic object with default properties and places it somewhere within the parent frame. Objects of following types can be created this way: `Arc`, `Ellipse`, `Group`, `GInset`, `GLine`, `Polyline`, `Polygon`, `Rectangle`, `RoundRect`, `TxFrame` (a new `Flow` object is created at the same time), `TxLine` and `UFrame`.

Arguments. One integer - code of object's subtype. For list of possible values see "Codes of object classes." on page 142. (Not all codes are applicable.)

Return value. The newly created object.

Exceptions. `TypeError` is generated when the argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker` and no object was created.

Binding. Bound to: `AFrame`, `UFrame`.

5.2.1.14 `Open()`

Description. Opens an existing book or document or creates a new document based on an existing one.

Arguments. Pathname of the file to open and dictionary of additional options that will influence the process of opening.

Return value. The just open `Book` or `Document` object or a new `Document` object.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`. Additional information is written to the variable `maker.ioStatus` in the form of list of strings.

Binding. Bound to: `Session`.

5.2.1.15 `paste()`

Description. Pastes the contents of the clipboard.

Arguments. Ignored.

Return value. Nothing.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `Document`.

5.2.1.16 `Print()`

Description. Prints book or document according to its current print settings.

Arguments. Ignored.

Return value. Nothing.

Exceptions. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `Book`, `Document`.

5.2.1.17 `Save()`

Description. Save (or export) an open book or document to a file.

Arguments. Pathname of the file to save and dictionary of additional options that will influence the process of saving.

Return value. Nothing.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`. Additional information is written to the

variable `maker.ioStatus` in the form of list of strings.

Binding. Bound to: `Book`, `Document`.

5.2.1.18 `select()`

Description. This is a way how to select ranges of table cells.

Arguments. Four integers: numbers of first and last row and first and last column of the new selected area. (Columns and rows are numbered from zero.) If the first parameter is set to `SelectWholeTable`, all remaining values are ignored and the whole table is selected.

Return value. Nothing.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `Table`.

5.2.1.19 `sOpen()`

Description. Opens an existing book or document using default settings.

Arguments. Takes two arguments: pathname of the file to open (string) and a number. If the number is non-zero, messages and warnings are displayed (including the open dialog box).

Return value. The just open `Book` or `Document` object.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker` and the file is not open.

Binding. Bound to: `Session`.

5.2.1.20 `sUpdate()`

Description. Updates the book with default settings and regenerates generated components.

Arguments. Two integers. When the first is non-zero, warnings, messages and error log will be

displayed. When the second is non-zero, generated files will be shown.

Return value. Nothing.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `Book`.

5.2.1.21 `text()`

Description. Text in `FrameMaker` is a sequence of formatted characters, mixed with special objects such as variables or tables, spread across layout units such as paragraphs, columns and frames. This method requests items describing all of these aspects you need for the object it is bound to.

Arguments. Variable number of integer parameters that symbolize type of information you need. For list of possible values see "Types of items that compose the text content of documents." on page 166.

Return value. A list of triples is returned. Every triple describes a single item. First member of a triple contains offset of the item from the begin of `Paragraph` or `TxLine` object containing it. Second member of a triple contains type of the item. This number equals to one of parameters specified. Third member of a triple contains additional information depending on item type - often a `makerObject` (with a few exceptions).

Exceptions. `TypeError` is generated when a non-integer parameter was specified. `RuntimeError` is generated when an error occurs in `FrameMaker`.

Binding. Bound to: `TbCell`, `Element`, `Flow`, `Footnote`, `Paragraph`, `TxColumn`, `TxFrame`, `TxLine`, `PlugInset`, `FInset`, `TxInset`, `TbInset`, `Variable`, `XRef`, `TxRange`.

5.2.1.22 `typeCode()`

Description. This is a way how to determine subtype of objects of type `makerObject`.

Arguments. Ignored.

Return value. Number. For list of possible values see "Codes of object classes." on page 142.

Exceptions. None.

Binding. Bound to: AFrame, Arc, BodyPage, Book, Component, TbCell, CharFormat, Color, CombiFont, Command, CondFormat, DlgResource, DlgBox, DlgButton, DlgCheckBox, DlgEditBox, DlgImage, DlgLabel, DlgPopUp, DlgRadioButton, DlgScrollBar, DlgScrollBox, DlgTriBox, Document, Element, ElementDef, Ellipse, Flow, FormatChange, FormatRule, FormatClause, Footnote, Group, HiddenPage, GInset, GLine, Marker, MarkerType, MasterPage, Expression, Menu, Separator, Paragraph, ParaFormat, Polygon, Polyline, Rectangle, RefPage, RoundRect, TbRow, Rubi, RulingFormat, Session, TxColumn, Table, TbFormat, TxFrame, TxLine, PlugInset, FInset, TxInset, TbInset, UFrame, Variable, VarFormat, XRef, XRefFormat.

5.2.1.23 Update()

Description. Updates the book with optional custom settings.

Arguments. Dictionary of additional options that will influence the update process. For list of possible values see "Options influencing updates of books." on page 169.

Return value. Nothing.

Exceptions. `TypeError` is generated when an argument is of incompatible type. `RuntimeError` is generated when an error occurs in `FrameMaker`. Additional information is written to the variable `maker.ioStatus` in the form of list of strings.

Binding. Bound to: Book.

Chapter 6

Symbolic values

There are numerous places across fmPython where symbolic values are used. Symbolic value is simply a number with a certain interpretation in some contexts and therefore it is more user friendly to give it a name that allows to guess the right meaning.

All symbolic values in fmPython are accessible through attributes of the object `symbol` in the module `maker`. You can even print their concrete values, if you wish, as in following example.

```
print maker.symbol.ClearFill
```

6.1 Fill patterns

Shapes of graphic objects are drawn and eventually filled using patterns. The image below contains examples for all possible patterns - each rectangle drawn on top of a cyanotic gray box is filled with pattern having the number shown. Patterns with numbers 0, 7 and 15 are somehow “special” (solid) and therefore have symbolic names. How these patterns will print depends on printer (driver). Patterns numbered 1 - 6 are true patterns when displayed by `FrameMaker`, but

for example by distilling to PDF they converted to solid colors.



6.2 Grouped symbols

Table 79: Types of anchoring. The frame will appear on the same page as its (text) anchor globally positioned accordingly to this type.

Name	Meaning
AnchorBelow	The frame is placed below the line of text containing its anchor.
AnchorBottom	The frame is placed as the last object bottom in the text frame. If you place more frames in the same frame this way, they can collide or overlap.
AnchorFrameFar	This type places the frame outside the text frame containing the anchor on the side farther from the page edge.
AnchorFrameInside	This type places the frame outside the text frame containing the anchor on the side closer to the page binding.
AnchorFrameLeft	This type places the frame on the left outside the text frame containing the anchor.

Name	Meaning
AnchorFrameNear	This type places the frame outside the text frame containing the anchor on the side closer to the page edge.
AnchorFrameOutside	This type places the frame outside the text frame containing the anchor on the side farther from the page binding.
AnchorFrameRight	This type places the frame on the right outside the text frame containing the anchor.
AnchorInline	This type is rather for smaller things as the frame is placed directly in the line and there is no anchor.
AnchorInPg	The <code>FrameMaker</code> will find an appropriate place within the paragraph containing the anchor. Paragraph's text will flow around.
AnchorSubcolFar	This type places the frame outside the text column containing the anchor on the side farther from the page edge.
AnchorSubcolInside	This type places the frame outside the text column containing the anchor on the side closer to the page binding.
AnchorSubcolLeft	This type places the frame on the left outside the text column containing the anchor.
AnchorSubcolNear	This type places the frame outside the text column containing the anchor on the side closer to the page edge.
AnchorSubcolOutside	This type places the frame outside the text column containing the anchor on the side farther from the page binding.
AnchorSubcolRight	This type places the frame on the right outside the text column containing the anchor.
AnchorTop	The frame is placed as the first object at the top of the text frame. If you place more frames in the same frame this way, they can collide or overlap.

Table 80: Options for displaying element attributes in structural view.

Name	Meaning
DisplayAll	All attributes of the element are displayed.
DisplayNone	No attribute of the element is displayed.
DisplayReqAndSpec	Only required and specified (having value) attributes of the element are displayed.

Table 81: Options for editing attributes of new structural elements.

Name	Meaning
AllwaysEdit	Attribute editor will start automatically for all new elements.
DoNotEdit	Attribute editor will not start automatically.
EditIfRequired	Attribute editor will start automatically if at last one attribute is required.

Table 82: Possibilities how to number paragraphs in tables.

Name	Meaning
AutonumByCols	Paragraphs in table cells are numbered column by column.
AutonumByRows	Paragraphs in table cells are numbered row by row.

Table 83: Types of background used for body pages.

Name	Meaning
CustomPageBG	The page has custom background specified by the master-Page attribute (on page 44).
DefaultPageBG	The page has a Left or Right master page background if the document is double-sided, or a Right master page background if the document is single-sided.
NoPageBG	The page has no master page background.

Table 84: When displaying letters there are these possibilities.

Name	Meaning
CaseLower	All characters are displayed lowercase.
CaseNormal	A mix of uppercase and lowercase characters is displayed.
CaseSmallCaps	All characters are uppercase, but the formerly lowercase are somewhat smaller.
CaseUpper	All characters are displayed uppercase.

Table 85: Element catalog display options (what elements are listed).

Name	Meaning
AllCatalog	All elements are listed (the complete catalog contents).
ChildCatalog	All elements valid at current structure level are listed.
CustomCatalog	Only elements specified by <code>customElements</code> attribute (on page 22) are listed.
LooseCatalog	Elements valid at current location in any order are listed.
StrictCatalog	Only elements valid at current location are listed.

Table 86: Options for positioning change bars.

Name	Meaning
ChangeBarFar	Change bars are placed on the side farthest from the page edge.
ChangeBarLeft	Change bars are placed to the left of the column.
ChangeBarNear	Change bars are placed on the side nearest to the page edge.
ChangeBarRight	Change bars are placed to the right of column.

Table 87: Codes of object classes.

Name	Meaning
AFrame	Anchored frame
Arc	Graphic arc
BodyPage	Body page
Book	Book
CharFormat	Character format
Color	Color
CombiFont	Combined font definition
Command	Command
Component	Book component
CondFormat	Condition format
DlgBox	Dialog box

Name	Meaning
DlgButton	Dialog button
DlgCheckBox	Check box
DlgEditBox	Edit box
DlgImage	Dialog image
DlgLabel	Dialog label
DlgPopUp	Pop-up menu
DlgRadioButton	Radio button
DlgResource	Dialog resource
DlgScrollBar	Scroll bar
DlgScrollBox	Scroll box
DlgTriBox	Three-state checkbox
Document	Document
Element	Structural element
ElementDef	Definition of structural element
Ellipse	Ellipse
Expression	Mathematic expression
FInset	Text imported from a flow
Flow	Text flow
Footnote	Footnote
FormatChange	Change of format
FormatClause	Formatting clause
FormatRule	Formatting rule
GInset	Imported graphic
GLine	Graphic line
Group	Group of objects
HiddenPage	Hidden page
Marker	Marker
MarkerType	Type of marker

Name	Meaning
MasterPage	Master page
Menu	Menu
ParaFormat	Paragraph format
Paragraph	Paragraph of text
PlugInset	Text imported by an extension module
Polygon	Polygon
Polyline	Polyline
Rectangle	Rectangle
RefPage	Reference page
RoundRect	Rectangle with rounded corners
Rubi	Rubi
RulingFormat	Format of ruling
Separator	Item separator in menu
Session	Frame product session
Table	Table
TbCell	Table cell
TbFormat	Format of table
TbInset	Table imported by reference
TbRow	Table row
TxColumn	Column of text
TxFrame	Text frame
TxInset	Text imported by reference
TxLine	Graphic text object
UFrame	Unanchored frame
VarFormat	Variable format
Variable	Variable
XRef	Cross-reference
XRefFormat	Cross-reference format

Table 88: Types of commands.

Name	Meaning
MenuItemAPI	The command is a menu item defined by a plug-in.
MenuItemExpand	The menu item is an expandomatic menu item (such as !ShowParagraphTags) defined by the FrameMaker product.
MenuItemFM	The command is a menu item defined by the FrameMaker product.
MenuItemMacro	The menu item is not a command - it calls a macro.

Table 89: What numbers will be given to numbered objects (such as paragraphs or pages) in a book component.

Name	Meaning
ContinueNum	Numbering will continue from the preceding component in book.
NumFromFile	Numbering as specified in the file will be used.
PerPageNum	Restart numbering on each page. This option applies to footnotes only.
RestartNum	Numbering will restart from an initial value.
SameNum	Use the same number as for previous component. This numbering applies only to objects possibly spanning multiple components, as chapters and volumes.

Table 90: Types of book components.

Name	Meaning
AuthorIdx	Generated index of authors.
FigureList	Generated list of figures.
FormatIdx	Generated index of formats.
FormatList	Generated list of formats.
MarkerIdx	Generated index of markers.
MarkerList	Generated list of markers.
MarkerListSorted	Generated list of markers, alphabetically sorted.

Name	Meaning
NotGenerated	Book component is not generated.
ParaList	Generated list of paragraphs.
ParaListSorted	Generated list of paragraphs, alphabetically sorted.
ReferenceIdx	Generated index of references.
ReferenceList	Generated list of references.
StandardIdx	Generated standard index.
SubjectIdx	Generated subject index.
TableList	Generated list of tables.
TOC	Generated table of contents.

Table 91: States of dialog controls.

Name	Meaning
ActiveState	This state does mean: for a button - used (clicked), for a check or three-state box - checked, for a radio button - on.
AsIsState	This state is valid for three-state boxes and commands to use „As Is” setting.
InactiveState	This state does mean: for a button - not used (clicked), for a check or three-state box - unchecked, for a radio button - off.

Table 92: Types of documents FrameMaker can open or save.

Name	Meaning
BinaryDocument	FrameMaker binary document.
FilteredDocument	Document filtered through a built-in or plug-in filter.
MIFDocument	MIF document (Maker Interchange Format).
TextDocument	ASCII text document.

Table 93: Options for displaying element bounds.

Name	Meaning
BracketBounds	Brackets are displayed at element bounds.

Name	Meaning
NoBounds	There are no visible bounds of elements in the text.
TaggedBounds	Start and end tags are displayed at element bounds.

Table 94: Types of structural elements.

Name	Meaning
CellElement	Element of this type represents one cell of a table.
ContainerElement	Element of this type represents a range in text flow.
FootnoteElement	Element of this type represents a footnote.
GraphicElement	Element of this type represents a graphic imported into an anchored frame.
MarkerElement	Element of this type represents a marker.
MathElement	Element of this type represents an expression (mathematic).
RowElement	Element of this type represents one row of a table.
RubiElement	Element of this type has something to do with asian characters.
RubiGroupElement	You should not get this symbolic value as there is no object corresponding with it.
TableBodyElement	Element of this type contains all body rows of a table.
TableElement	Element of this type represents a table.
TableFooterElement	Element of this type contains all footer rows of a table.
TableHeaderElement	Element of this type contains all header rows of a table.
TableTitleElement	Element of this type represents the title that appears above or below all parts of a table (if the table is distributed across more columns/frames/pages).
UnspecifiedElement	I never got this element type - and you?
VariableElement	Element of this type represents a variable.
XrefElement	Element of this type represents a cross-reference.

Table 95: Styles available for footnote autonumbering.

Name	Meaning
CustomNumbering	Characters from the <code>footnoteChars</code> attribute (on page 27) are used.
DaijiNumbering	Daiji is used.
FnoteNumAlphaLC	Latin letters (a to z) are used.
FnoteNumAlphaUC	Latin letters (A to Z) are used.
FnoteNumArabic	Arabic digits (0 to 9) are used.
FnoteNumKanji	Kanji is used.
FnoteNumRomanLC	Roman digits (i, v, x, l, c, d, m) are used.
FnoteNumRomanUC	Roman digits (I, V, X, L, C, D, M) are used.
FnoteNumZenkaku	Zenkaku is used.
FnoteNumZenkakuLC	Zenkaku in lowercase is used.
FnoteNumZenkakuUC	Zenkaku in uppercase is used.
KazuNumbering	Kazu is used.

Table 96: Horizontal alignment options of anchored frames. (Not all of them are valid for every type of anchoring.)

Name	Meaning
FrameAlignCenter	The frame appears centered within text frame.
FrameAlignInside	The frame will be placed on the side closer to the binding of pages.
FrameAlignLeft	The frame appears on the left side of the text frame.
FrameAlignOutside	The frame will be placed on the side farther from the binding of pages.
FrameAlignRight	The frame appears on the right side of the text frame.

Table 97: Possible values to which `hypertextCommandCode` attribute (on page 71) can be set.

Name	Meaning
CmdAlert	alert command

Name	Meaning
CmdAlertTitle	alerttitle command
CmdError	Parser is in an error state
CmdExit	exit commant
CmdGoLink	gotolink command
CmdGoLinkFit	gotolinkfitwin command
CmdGoNew	gotonew command
CmdGoObject	gotoObjectId command
CmdGoObjectFit	gotoObjectIdfitwin command
CmdGoPage	gotopage command
CmdMatrix	matrix command
CmdMessage	message command
CmdNewLink	newlink command
CmdNextPage	nextpage command
CmdNoOp	Command causes no event
CmdOpenLink	openlink command
CmdOpenLinkFit	openlinkfitwin command
CmdOpenNew	opennew command
CmdOpenObject	openObjectId command
CmdOpenObjectFit	openObjectIdfitwin command
CmdOpenPage	openpage command
CmdPopUp	popup command
CmdPrevLink	previouslink command
CmdPrevLinkFit	previouslinkfitwin command
CmdPrevPage	previouspage command
CmdQuit	quit command
CmdQuitAll	quitall command
CmdUnknown	Unknown command

Table 98: Possible values to which `hypertextDestType` attribute (on page 71) can be set.

Name	Meaning
DestObjDataLink	Destination object is subscribed data.
DestObjElement	Destination object is an element.
DestObjGraphic	Destination object is a graphic.
DestObjMarker	Destination object is a marker.
DestObjPgf	Destination object is a paragraph.
DestObjTextInset	Destination object is a text inset.
DestObjUnknown	Unknown or invalid object as destination.
DestObjXRef	Destination object is a cross-reference.

Table 99: Possible values to which `hypertextDestination` attribute (on page 71) can be set.

Name	Meaning
DestFirstPage	Destination is the first page of a file
DestFluidFlow	Destination is to a fluid flow document
DestLastPage	Destination is the last page of a file
DestMarker	Destination is a marker
DestNewLink	Destination is a newlink
DestObject	Destination is an object ID (usually for generated hypertext commands)
DestPageNum	Destination is a named page (usually a page number)
DestXRef	Destination is a cross-reference
NoDestination	No destination found

Table 100: Possible error codes to which `hypertextError` attribute (on page 71) can be set.

Name	Meaning
HTBadCmd	Cannot map the first keyword to an existing <code>hypertext-CommandCode</code> attribute (on page 71).

Name	Meaning
HTBadPath	File reference expected for this command, but no valid filepath found.
HTEmptyCmd	Hypertext string is empty.
HTExtraArgs	More than the required number of arguments for the command; extra arguments were ignored.
HTMissingArgs	One or more arguments required for the command is missing.
HTNoHelp	Default help directory either does not exist (help was not installed) or cannot be found.
HTNumberExpected	Command expected a number but got text; check <code>hypertextErrorIndex</code> attribute (on page 71).
HTSyntaxOK	No parse errors.
HTUnrelatedPath	File reference is relative to the current document, but the current document has not been saved; file location could not be calculated.

Table 101: Possible error codes to which `hypertextValidateErr` attribute (on page 73) can be set.

Name	Meaning
HTBadDimension	One or both of the matrix dimensions is bad; must be between 1 and 99.
HTBadObject	The referenced file is valid, but the link is to an object with an unrecognized object type.
HTCantOpen	Can't open the file; perhaps you don't have permission, or the file is locked.
HTIllegalCmd	Invalid command in the popup command's reference flow; for example, matrix or newlink.
HTInvalidCmd	One of the commands in the reference page flow for a matrix command has a parse or validation error.
HTInvalidCode	Invalid F-code in the hypertext command.
HTMissingCmd	At least one entry in the popup command's reference flow has no text in it.
HTMissingFlow	Can't find the named reference flow for a matrix or popup command.

Name	Meaning
HTMissingLabel	At least one entry in the popup command's reference flow has no text in it.
HTMissingLine	The reference flow for a matrix or popup command is missing one or more lines.
HTMissingLink	The referenced file is valid, but can't find the named link within it.
HTMissingMarker	At least one entry in the popup command's reference flow has no hypertext marker in it.
HTMissingObject	A link to an object, but can't find the object.
HTMissingPage	The referenced file is valid, but can't find the specified page.
HTNotDoc	The referenced file is not a valid <code>FrameMaker</code> document.
HTNotFile	The referenced file could not be found, or is not a regular file; for example, it could be a directory name.
HTRecursion	The reference flow for a matrix or popup command contains nested popup or matrix commands that name a parent reference flow.
HTUsesDefault	Warning: default text was found as an argument; maybe you forgot to replace it by actual value.
HTValid	No validation errors.

Table 102: Options influencing importing operations.

Name	Meaning
CellSeparator	If <code>TextFile</code> is <code>ImportTable</code> , the delimiter or separator (string) used to parse the text into cells.
DisallowGraphics	Whether to disable importing of graphics. Possible values: <code>false</code> - graphics will be imported (default), <code>true</code> - don't import such files.
DisallowMac	Whether to disable importing of/from Macintosh documents. Possible values: <code>false</code> - allow import of/from Macintosh editions (default), <code>true</code> - don't allow import of/from such files.
DocumentFile	What to do when the file to import is <code>FrameMaker</code> document (binary <code>.fm</code> or <code>.mif</code>). Possible values: <code>CancelOperation</code> , <code>ContinueOperation</code> (default), <code>ShowDialog</code> .

Name	Meaning
EmptyHeading	Whether to leave table heading rows empty when <code>TextFile</code> is set to <code>ImportTable</code> . Possible values: <code>false</code> - fill them (default), <code>true</code> - leave them empty.
FitInSelectedRect	Whether to fit the graphic within the selected frame. Possible values: <code>true</code> - fit (default), <code>false</code> - do not fit. However, there is probably no possibility to programmatically select a frame (or another graphic object).
GraphicDPI	Resolution at which to import the graphic, if applicable, in dots per inch (default: 72).
GraphicFile	What to do when the file to import is graphic. Possible values: <code>CancelOperation</code> , <code>ContinueOperation</code> (default), <code>ShowDialog</code> .
HeadingRows	Number of heading rows in the table when <code>TextFile</code> is set to <code>ImportTable</code> . Default value: 1.
HowToImport	How to import the file. Possible values: <code>ImportByRef</code> (default), <code>ImportByCopy</code> , <code>UserDecision</code> .
ImportApplication	Name of SGML application to use when importing SGML document with <code>FrameMaker+SGML</code> .
ImportFlow	If <code>MainFlow</code> is <code>false</code> , the name of the flow to import.
ImportText	Whether to import the file as text even if its type is different (MIF, filterable, ...). Possible values: <code>false</code> - import it according to its type, <code>true</code> - import it as text.
ImportType	Specifies format of the file to import. Possible values: <code>RecognizeType</code> (default), <code>BinaryType</code> , <code>MIFType</code> , <code>TextType</code> , <code>SGMLType</code> , <code>FilterType</code> .
InsetName	Name of text inset resulting from the import operation.
LineIsRow	How to convert text lines when <code>TextFile</code> is set to <code>ImportTable</code> . Possible values: <code>true</code> - convert each text line to one table row using <code>CellSeparator</code> and <code>NumSeparators</code> to determine cells, <code>false</code> - convert every text line into a cell.
MainFlow	Whether to import the main flow from the document. Possible values: <code>true</code> - import it (default), <code>false</code> - import flow specified by <code>ImportFlow</code> and <code>PageSpace</code> .
ManualUpdate	How to update the resulting inset. Possible values: <code>false</code> - autoupdate (default), <code>true</code> - update manually.

Name	Meaning
NumColumns	Number of columns in the table when LineIsRow is set to false and TextFile is set to ImportTable. Default value: 1.
NumSeparators	Number of spaces used as cell separator when CellSeparator is set to space and TextFile is set to ImportTable. Default value: 1.
PageSpace	If MainFlow is false, the type of pages to search for the flow specified by ImportFlow. Possible values: BodySpace (default), ReferenceSpace.
RemoveOverrides	Whether to remove imported format overrides if TextFormat is set to TargetFormat. Possible values: true - remove them (default), false - don't touch them.
RemovePageBreaks	Whether to remove imported manual page breaks if TextFormat is set to TargetFormat. Possible values: true - remove them (default), false - don't touch them.
SGMLFile	What to do with SGML documents. Possible values: ContinueOperation (default), CancelOperation, ShowDialog.
TableFormat	Name of table format to use when TextFile is set to ImportTable.
TextFormat	How to format the imported text. Possible values: TargetFormat (default), PlainFormat, SourceFormat.

Table 103: Options influencing operations of import and open.

Name	Meaning
DisallowFiltered	Whether to disable handling of filterable files. Possible values: false - filterable file will be handled (default), true - don't handle such files.
DisallowMIFDoc	Whether to disable handling of Frame documents (MIF form). Possible values: false - MIF document (.mif) will be handled (default), true - don't handle such documents.
DisallowSGML	Whether to disable handling of SGML files. Possible values: false - SGML files will be handled (default), true - don't handle such files.

Name	Meaning
DisallowText	Whether to disable handling of plain text documents. Possible values: <code>false</code> - plain text files will be handled (default), <code>true</code> - don't handle such files.
FileTypeHint	For filterable files the string that enables <code>FrameMaker</code> to use the correct filter.
ShowBrowser	Whether to show the dialog box for file operations. Possible values: <code>false</code> - don't display the dialog (default), <code>true</code> - display it.
TextFile	How to handle text files. Possible values: <code>EOLisEOP</code> (default), <code>EOLisntEOP</code> , <code>ShowDialog</code> , <code>CancelOperation</code> , <code>ImportTable</code> - for import only.

Table 104: Options influencing operations of import, open and save.

Name	Meaning
AlertUser	What to do when unexpected conditions occur. Possible values: <code>false</code> - don't display alerts (default), <code>true</code> - notify user.
DontNotifyOthers	Whether to notify other API clients (plug-ins) about a successful operation. Possible values: <code>false</code> - other plug-ins will be notified (if they requested it, default), <code>true</code> - other plug-ins will NOT be notified.

Table 105: Currently unclassified symbolic values.

Name	Meaning
AskName	Prompt user for filename.
BinaryType	<code>FrameMaker</code> binary book or document (<code>.book</code> , <code>.fm</code>).
BodySpace	Search for flow on body pages.
CancelOperation	Cancel the operation.
CellsByColumns	Save each cell as a paragraph column-by-column.
CellsByRows	Save each cell as a paragraph row-by-row.
ContinueOperation	It is OK to complete the operation.
CurrentPages	Save all current pages.

Name	Meaning
CurrentSetting	Use option as set in the document by <code>pageRounding</code> attribute (on page 76).
DoBackup	Backup previous version.
DoNotPerform	Do not perform the operation (update etc.).
EmptyNotSaved	Empty pages at document end will not be saved.
EmulsionDown	Print emulsion side down.
EmulsionUp	Print emulsion side up.
EndOffset	Offset after the last object in a text object. This value can be used to reference locations relatively to the end of the text object.
EOLisEOP	Convert ends of lines to paragraph breaks when opening text file.
EOLisntEOP	Do not convert ends of lines to paragraph breaks when opening text file.
FilterType	Use filter to handle this type of file. It is necessary to specify the <code>FileTypeHint</code> , too.
ImportByCopy	Embed file's data directly into document.
ImportByRef	Import file by reference.
ImportTable	Import text file as table.
LastSettings	Use settings as lastly defined in appropriate dialog.
MIFType	Book or document in Maker Interchange Format (.mif).
ModeSave	The file should be saved under its original name.
ModeSaveAs	The file should become a new name and <code>SaveNameMode</code> option comes into play.
NoBackup	Make no backup.
OpenCopy	Force open of an editable copy of book or document.
OpenViewOnly	Open view-only copy of book or document.
Perform	Perform the operation (update etc.).
PreferenceBackup	When considering backup follow what's specified by <code>auto-Backup</code> attribute (on page 108).
PrintAll	Print all pages of the document.

Name	Meaning
PrintRange	Print only a range of pages.
ProvidedName	Save under the name provided to the save method.
RecognizeType	Recognize type of the file automatically.
ReferenceSpace	Search for flow on reference pages.
ResetLock	Reset lock and open the file.
SaveEvenPage	When the number of pages is odd add an (empty) extra page at end.
SaveFiltered	Save through a filter. Set <code>SaveTypeHint</code> to let <code>FrameMaker</code> determine the filter. (HTML and XML are saved via filters, too.)
SaveFM	Save in native format of <code>FrameMaker</code> (.fm, .book).
SaveMIF	Save in Maker Interchange Format (.mif).
SaveOddPage	When the number of pages is even add an (empty) extra page at end.
SavePdf	Print to PostScript and invoke Acrobat Distiller to create PDF.
SaveSgml	Save in SGML format - <code>FrameMaker</code> +SGML only.
SaveStationery	Save in stationery (?) format - Macintosh only.
SaveText	Save only the text.
SaveViewOnly	Save in view-only format.
SGMLType	Structured SGML document (.sgm, .sgml).
ShowDialog	Display dialog box and let the user decide.
SkipTables	Skip tables when saving as text.
StartSideFromFile	Book component begins on the side as specified in the file.
StartSideLeft	Book component begins on left side.
StartSideNext	Book component begins on next available side.
StartSideRight	Book component begins on right side.
StripStructure	Remove structure from the book or document and open it.
TextType	File containing ordinary text (.txt).
TitleName	Use the string appearing in title bar of document's window as filename.

Name	Meaning
UserDecision	Display dialog box and let the user decide.
UserPreference	Handle the file according to its internal setting.

Table 106: Line of text can have the following types of end. (An ordinary line end has type of zero.)

Name	Meaning
HardLineEnd	Line end is forced.
HyphenLineEnd	A word at line end is broken and therefore the line ends with a hyphen.

Table 107: Options how to compute height of lines in text.

Name	Meaning
LinesFixed	Line height is fixed to the default paragraph font.
LinesFloating	Line height is dependent on largest ascender.
LinesProportional	Line height is dependent on largest font.

Table 108: Values `enabledWhen` attribute (on page 55) can have and the corresponding contexts in which a menu item is active.

Name	Meaning
AlwaysDisable	Disabled in all contexts. If a menu item is enabled and you set <code>enabledWhen</code> attribute (on page 55) to this value, it disables and dims the menu item.
AlwaysEnable	Enabled in all contexts. This is the default value. If the menu item is disabled, setting <code>enabledWhen</code> attribute (on page 55) to this value allways enables it.
EnIfAFrameSelected	The first selected object is an anchored frame.
EnIfCanCopy	Some text or an object is selected.
EnIfCanCopyFont	The insertion point or selection is in the text of a paragraph, an expression, a table, or a text line.
EnIfCanPaste	The Clipboard contains an object or text that can be pasted at the insertion point.

Name	Meaning
EnIfCellSelected	A single cell in a table is selected.
EnIfCellsSelected	One or more cells in a table are selected.
EnIfGInsetSelected	The first selected object is a graphic inset.
EnIfObjectSelected	An object is selected.
EnIfObjProps	The insertion point is in text, a table, or an expression, or a graphic object is selected.
EnIfOrInFrameSelected	The selected object is a graphic frame or is in a graphic frame that is not a page frame.
EnIfPgfSelection	The selection is in a paragraph.
EnIfTableSelected	An entire table is selected.
EnIfTextFrameSelected	A text frame is selected.
EnIfTextInsetSelected	The first selected object is a text inset.
EnInBook	A book is open.
EnInCell	The insertion point or selection is in a table cell.
EnInDoc	A document is open.
EnInExpression	The insertion point or selection is in an expression.
EnInFlow	A text frame is selected, or the insertion point or selection is in a paragraph.
EnInPgf	The insertion point or selection is in a paragraph (but not in an expression).
EnInTable	The insertion point or selection is in any part of a table.
EnInTableTitle	The insertion point or selection is in the table title.
EnInText	The insertion point or selection is in a graphic text line or a paragraph.
EnInTextLine	The insertion point or selection is in a graphic text line.
EnInViewOnly	The current document is locked.

Table 109: Types of menu.

Name	Meaning
MenuAdHocRuler	An ad hoc formatting menu that appears on the ruler.

Name	Meaning
MenuBarFM	A menu bar defined by the FrameMaker product.
MenuDefault	A pull-down or pull-right menu.
MenuPopUp	A pop-up menu.

Table 110: Symbolic measurement units you can use when working with metric values.

Name	Meaning
cc	cicero (1 cc = 12 dd)
cm	centimeter (2.54 cm = 1 in)
dd	didot
in	inch (1 in = 72 pt)
mm	millimeter (25.4 mm = 1 in)
pc	pica (1 pc = 12 pt)
pt	point (72 pt = 1 in)

Table 111: Styles available for autonumbering.

Name	Meaning
NumAlphaLC	Latin letters (a to z) are used.
NumAlphaUC	Latin letters (A to Z) are used.
NumArabic	Arabic digits (0 to 9) are used.
NumDaiji	Daiji is used.
NumKanji	Kanji is used.
NumKanjiKazu	Kazu is used.
NumRomanLC	Roman digits (i, v, x, l, c, d, m) are used.
NumRomanUC	Roman digits (I, V, X, L, C, D, M) are used.
NumText	Text is used. This style is available for volumes and chapters only.
NumZenkaku	Zenkaku is used.
NumZenkakuLC	Zenkaku in lowercase is used.

Name	Meaning
NumZenkakuUC	Zenkaku in uppercase is used.

Table 112: Options influencing opening operations.

Name	Meaning
BeefyDocument	On Macintosh platform there could be a problem when opening big documents. This option specifies what to do in such cases. Possible values: CancelOperation, ContinueOperation, ShowDialog.
BookLocked	What to do when the book to open is already open (there is a lock to the book file). Possible values: OpenViewOnly (default), OpenCopy, ShowDialog, ResetLock, CancelOperation.
BookViewOnly	Whether to open a book for viewing only. Possible values: false - normal open based on the file (default), true - open book for viewing only.
CreateNew	Whether to open the document or create a new one based on it. Possible values: false - just open the document (default), true - create a new document based on the original.
DisallowFMBook	Whether to disable opening of Frame books (binary form). Possible values: false - binary Frame books (.book) will be opened (default), true - don't open such books.
DisallowFMDoc	Whether to disable handling of Frame documents (binary form). Possible values: false - binary Frame document (.fm) will be handled (default), true - don't handle such documents.
DisallowMIFBook	Whether to disable opening of Frame books (MIF form). Possible values: false - binary Frame books (.mif) will be converted and opened (default), true - don't open such books.
DocViewOnly	Whether to open a document for viewing only. Possible values: false - normal open based on the file (default), true - open document for viewing only.
FontMetricDiffers	What to do when a font metric information stored in the document differs from that of current session. Possible values: CancelOperation (default), ContinueOperation, ShowDialog.
IgnoreReferences	Open file even if its references can't be resolved.

Name	Meaning
MakeIconic	Whether to iconize the just open book's or document's window (UNIX and Windows only). Possible values: <code>false</code> - leave an ordinary window, <code>true</code> - make an iconic window.
MakeVisible	Whether to make a window for the just open book or document. Possible values: <code>false</code> - make no window, <code>true</code> - make a window (default). (FrameMaker opens documents this way for example when updating books.)
MissingCatalogFont	What to do when the format catalog of the document to open references unavailable fonts. Possible values: <code>CancelOperation</code> (default), <code>ContinueOperation</code> , <code>ShowDialog</code> .
MissingFont	What to do when the document to open uses unavailable fonts. Possible values: <code>CancelOperation</code> (default), <code>ContinueOperation</code> , <code>ShowDialog</code> .
MissingLanguage	What to do when the document to open uses unavailable languages. Possible values: <code>CancelOperation</code> (default), <code>ContinueOperation</code> , <code>ShowDialog</code> .
MissingReference	What to do when the file to open imports another file(s) by reference but these are unavailable. Possible values: <code>CancelOperation</code> (default), <code>IgnoreReferences</code> , <code>ShowDialog</code> .
NewWindow	Whether to open file in a new window. Possible values: <code>true</code> - file will be open in a new window (default), <code>false</code> - reuse window of the document specified by <code>OpenId</code> .
OldVersion	What to do when the book or document to open was created by a previous version of FrameMaker. Possible values: <code>CancelOperation</code> (default), <code>ContinueOperation</code> , <code>ShowDialog</code> .
OpenApplication	Name of SGML application to use when opening SGML document with FrameMaker+SGML.
OpenId	Document in window of which to open the document if <code>NewWindow</code> is set to <code>false</code> .
OpenText	Whether to open the file as text even if its type is different (MIF, filterable, ...). Possible values: <code>false</code> - open it according to its type, <code>true</code> - open it as text.
OpenType	Specifies format of the file to open. Possible values: <code>RecognizeType</code> (default), <code>BinaryType</code> , <code>MIFType</code> , <code>TextType</code> , <code>SGMLType</code> , <code>FilterType</code> .

Name	Meaning
StructuredFile	What to do when the book or document is structured but the current FrameMaker product does not support structure. Possible values: OpenViewOnly (default), CancelOperation, StripStructure, ShowDialog.
TitleString	String that will appear on the title bar of the open document's window (does not show on Macintosh).
UnwritableOpen	What to do when the file to open is not writable. Possible values: CancelOperation (default), ContinueOperation, ShowDialog.
UpdateDirectory	Whether to update directory of the dialog box for file operations. Possible values: false - don't update the directory (default), true - update it.
UpdateTextRefs	Specifies format of the file to import. Possible values: RecognizeType (default), BinaryType, MIFType, TextType, SGMLType, FilterType.
UpdateXRefs	What to do with crossreferences - update or not? Possible values: UserPreference - handle according to dontUpdateXRefs attribute (on page 66) (default), Perform, DoNotPerform.
UseAutosaved	What to do when there is an autosaved version of the file to open. Possible values: CancelOperation (default), Perform - use it, DoNotPerform - don't use it, ShowDialog.
UseRecovery	What to do when there is a recovery version of the file to open. Possible values: CancelOperation (default), Perform - use it, DoNotPerform - don't use it, ShowDialog.

Table 113: Options influencing operations of open and save.

Name	Meaning
DocumentLocked	What to do when the document to handle is already open from elsewhere (there is a lock on the document file). Possible values: OpenViewOnly (default when opening), OpenCopy, ShowDialog, ResetLock, CancelOperation (default when saving).
HardLocked	What to do when the book or document to handle is locked and the lock cannot be reset. Possible values: CancelOperation (default), ContinueOperation, ShowDialog.

Table 114: Options for page count rounding.

Name	Meaning
DeleteEmptyPages	Delete empty pages at end.
PageCountEven	Make page count even (may add empty pages at end).
PageCountOdd	Make page count odd (may add empty pages at end).
PageCountOriginal	Don't change page count.

Table 115: Horizontal placement options for paragraphs.

Name	Meaning
PgfAcrossColumns	\N
PgfAcrossFrame	\N
PgfInColumn	Paragraph is placed in the body subcolumn of the column.
RunInHead	\N
SideHeadAlignFirstBase	\N
SideHeadAlignLastBase	\N
SideHeadAlignTop	\N

Table 116: Vertical placement options for paragraphs.

Name	Meaning
PgfAtTopOfColumn	Paragraph is moved to top of (next) column, if it is not already there.
PgfAtTopOfLeftPage	Paragraph is moved to top of (next) left page, if it is not already there.
PgfAtTopOfPage	Paragraph is moved to top of (next) page, if it is not already there.
PgfAtTopOfRightPage	Paragraph is moved to top of (next) right page, if it is not already there.
PlacePgfAnywhere	Paragraph simply follows its predecessor.

Table 117: Options influencing save operations.

Name	Meaning
BlankLineEOP	Whether to add an extra (empty) line at ends of paragraphs when SaveType is set to SaveText. Possible values: false - write paragraphs as (long) lines and don't add anything, true - write lines as they are and an extra empty line at ends of paragraphs.
DateChanged	Somebody else has touched the file to be saved during current session, so what to do? Possible values: CancelOperation (default), ShowDialog, ContinueOperation.
DefaultPermissions	Whether to use the default UNIX permissions for writing. Possible values: false - use UNIXpermissions, true (default) - use current permissions of the file; when it is new, use RW for all categories (0666).
NotWritable	What to do when operating system does not permit to write to the file. Possible values: CancelOperation (default), ShowDialog.
SaveApplication	Name of SGML application to use when saving structured document as SGML with FrameMaker+SGML.
SaveBackup	Whether to backup previous version of the file. Possible values: PreferenceBackup (default), DoBackup, NoBackup.
SaveMode	What save mode to use. Possible values: ModeSaveAs (default), ModeSave.
SaveNameMode	What name to use when SaveMode is set to ModeSaveAs. Possible values: ProvidedName (default), TitleName, AskName.
SavePageCount	How to round the page count. Possible values: CurrentSetting (default), CurrentPages, SaveEvenPage, SaveOddPage, EmptyNotSaved.
SaveTextDialog	Whether to show the dialog box for text writing. Possible values: false - don't display it (default), true - display the dialog.
SaveTextTables	How to handle tables when SaveType is set to SaveText. Possible values: LastSettings (default), CellsByRows, CellsByColumns.

Name	Meaning
SaveTitleString	Whether to save the string appearing on title bar of document's window. Possible values: <code>false</code> - replace the string with file-name (default), <code>true</code> - don't change the string.
SaveType	Type of file to save to. Possible values: <code>SaveFM</code> (default), <code>SaveMIF</code> , <code>SaveStationery</code> , <code>SaveViewOnly</code> , <code>SaveSgml</code> , <code>SaveText</code> , <code>SavePdf</code> , <code>SaveFiltered</code> .
SaveTypeHint	This string tells <code>FrameMaker</code> what filter to use when <code>SaveType</code> is set to <code>SaveFiltered</code> . To save HTML use „0001ADBEHTML” and to save XML use „0001ADBEXML”.
UNIXpermissions	Value specifying UNIX permissions when <code>DefaultPermissions</code> is set to <code>false</code> . The value is best readable in octal form.
UpdateRecentList	Whether the file should be added to the list of recently visited in File menu. Possible values: <code>false</code> - don't add it (default), <code>true</code> - add it to the list.

Table 118: Options how to display components in a book.

Name	Meaning
AsFile	Pathname of the book component is displayed.
AsText	Text contents of the first paragraph of the component's main flow or value of the <code>displayText</code> attribute (on page 45) is displayed.

Table 119: Types of items that compose the text content of documents.

Name	Meaning
AFrameItem	This item represents an anchored frame.
ElementBeginItem	Container structural element begins at this location. All that follows this item is placed one level deeper in the structure.
ElementEndItem	Container structural element ends at this location. All that follows this item is placed one level less deep in the structure.
FlowBeginItem	Marks place where a flow begins. However, you can get probably at most one item of this type for all queried objects, and it will be first in the resulting list, if ever.

Name	Meaning
FlowEndItem	Marks place where a flow ends. However, you can get probably at most one item of this type for all queried objects, and it will be last in the resulting list, if ever.
FootnoteItem	This item represents a footnote.
FormatChangeItem	Text formatting properties change at this location. The additional information is a number whose bits identify what properties changed.
FrameBeginItem	Specifies text frame the following items are in (until the nearest frame end).
FrameEndItem	Specifies text frame the preceding items are in (from the nearest frame begin).
InsetBeginItem	Marks location where a text imported by reference or plug-in begins. Following items (until the corresponding end item) cannot be changed directly.
InsetEndItem	Marks location where a text imported by reference or plug-in ends. Preceding items (from the corresponding begin item) cannot be changed directly.
LineBeginItem	Marks location where a single line of text begins. The additional information in this case is meaningless and it will be probably zero.
LineEndItem	Marks location where a single line of text ends. The additional information describes type of line end. The value of zero describes a natural line end. For other possibilities For list of possible values see "Line of text can have the following types of end. (An ordinary line end has type of zero.)" on page 158.
MarkerItem	This item represents a marker.
PageBeginItem	Specifies page the following items are on (until the nearest page end).
PageEndItem	Specifies page the preceding items are on (from the nearest page begin).
ParaBeginItem	Marks location where a paragraph begins.
ParaEndItem	Marks location where a paragraph ends.
PrefixBeginItem	Prefix of an structural element, if it has declared one, begins at this location.

Name	Meaning
PrefixEndItem	Prefix of an structural element, if it has declared one, ends at this location.
StringItem	String of characters having all the same formatting properties and conditions applied. The additional information is a string.
SubColBeginItem	Specifies text column the following items are in (until the nearest column end).
SubColEndItem	Specifies text column the preceding items are in (from the nearest column begin).
SuffixBeginItem	Suffix of an structural element, if it has declared one, begins at this location.
SuffixEndItem	Suffix of an structural element, if it has declared one, ends at this location.
TableItem	This item represents a table. The table is anchored at its location.
TextObjItem	The object the offsets of following items are relative to. It can be a Paragraph or TxLine object.
VarBeginItem	Text variable begins here.
VarEndItem	Text variable ends here.
XRefBeginItem	Cross reference begins here.
XRefEndItem	Cross reference ends here.

Table 120: Options for formatting texts imported by reference.

Name	Meaning
PlainFormat	The text is formatted as plain text.
SourceFormat	The text is formatted with formats from the source document.
TargetFormat	The text is formatted with formats from the document into which it is imported.

Table 121: Options where to display table title.

Name	Meaning
NoTitle	The table has no title.

Name	Meaning
TitleAbove	The title appears above the table.
TitleBelow	The title appears below the table.

Table 122: Options influencing updates of books.

Name	Meaning
AlienFiles	What to do when there are files not created by FrameMaker in the book. Possible values: CancelOperation, ContinueOperation (default), ShowDialog.
InconsistentNumbering	What to do when numbering settings in book and its documents differ. Possible values: CancelOperation, ContinueOperation (default), ShowDialog.
ShowLog	Whether to show error log for this update operation. Possible values: true - show log, false (default) - do not create and error log, display alerts instead.
UpdateGenerated	Whether to update generated components of the book. Possible values: true - update (default), false - don't update.
UpdateNumbering	Whether to update numbering in all components of the book. Possible values: true - update (default), false - don't update.
UpdateOle	Whether to update OLE-links in all components of the book. Possible values: true - update (default), false - don't update.
UpdateReferences	Whether to update referenced texts (text insets) in all components of the book. Possible values: true - update (default), false - don't update.
UpdateXReferences	Whether to update cross-references in all components of the book. Possible values: true - update (default), false - don't update.
ViewerFiles	What to do when there are files for viewing only in the book. Possible values: CancelOperation, ContinueOperation (default) - this allows to modify viewer documents(!), ShowDialog.

Table 123: Selection options in view only mode.

Name	Meaning
ViewOnlyCantSelect	The user can't select text, and links targets do not highlight.
ViewOnlySelectAll	The user can select text (using modifier keys) and link targets are highlighted.
ViewOnlySelectUser	The user can select text when pressing modifier keys, and link targets (cross-reference sources and newlinks) do not highlight.

Table 124: Cross-reference options in view only mode.

Name	Meaning
ExternalXRefsActive	External cross-references are active.
InternalXRefsActive	Internal cross-references are active.
XRefsAlerts	An alert appears when cross-reference is clicked.
XRefsNotActive	Cross-references are not active.