

E l m S c r i p t

Version 7.0

**C o m m a n d
a n d
O b j e c t
R e f e r e n c e**

ELMSOFT INC.

7954 Helmart Drive

Laurel Maryland 20723

USA

Copyright © 1997-2011 ElmSoft, Inc. All rights reserved.

ElmSoft, Inc. ("ElmSoft") and its licensors retain all ownership rights to the ElmScript computer program and other computer programs offered by ElmSoft (hereinafter collectively called "ElmSoft Software") and their documentation. Use of ElmSoft Software is governed by the license agreement accompanying your original media. The ElmSoft Software source code is a confidential trade secret of ElmSoft. You may not attempt to decipher, decompile, develop, or otherwise reverse engineer ElmSoft Software, or knowingly allow others to do so. Information necessary to achieve the interoperability of the ElmSoft Software with other programs may be available from ElmSoft upon request. You may not develop passwords or codes or otherwise bypass the security features of ElmSoft Software.

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

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

Please remember that existing artwork or images that you may desire to scan as a template for your new image may be protected under copyright law. The unauthorized incorporation of such artwork or images into your new work could be a violation of the rights of the author. Please be sure to obtain any permission required from such authors.

ElmScript and ElmSoft are trademarks of ElmSoft.

Adobe, the Adobe logo, Acrobat, Acrobat Exchange, Adobe Type Manager, ATM, Display PostScript, Distiller, Exchange, Frame, FrameMaker, FrameMaker+SGML, FrameMath, FrameReader, FrameViewer, FrameViewer Retrieval Tools, Guided Editing, InstantView, PostScript, and SuperATM are trademarks of Adobe.

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

The following are copyrights of their respective companies or organizations:

The following are trademarks or registered trademarks of their respective companies or organizations:

Apple, AppleLink, AppleScript, AppleTalk, Balloon Help, Finder, ImageWriter, LaserWriter, PowerBook, QuickDraw, QuickTime, TrueType, XTND System and Filters, Macintosh, and Power Macintosh are used under license / Apple Computer, Inc.

Microsoft, MS-DOS, Windows / Microsoft Corporation

Sun Microsystems, Sun Workstation, TOPS, NeWS, NeWSprint, OpenWindows, SunView, SunOS, NFS, Sun-3, Sun-4, Sun386i, SPARC, SPARCstation / Sun Microsystems, Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Written and designed at Elmsoft, Inc., 7954 Helmart Drive, Laurel, MD 20723, USA

For civilian agencies: Restricted Rights Legend. Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in ElmSoft's standard commercial agreements for this software. Unpublished rights reserved under the copyright laws of the United States. The contractor/manufacturer is Elmsoft, Inc., 7954 Helmart Drive, Laurel, MD 20723, USA.

Table of Contents

1 Command Reference	1
Command Format	1
Add Commands	1
Add CommandObject	1
Add Member	2
Add MenuObject	3
Add MenuSepObject	3
Add Property	4
Apply Commands	5
Apply AttrExpression	5
Apply BookConditionalBuildExpressions	6
Apply BookConditionalSettings	6
-----	7
Apply PageLayout	7
Apply TextProperties	8
CallClient	9
Clear	11
Clear ChangeBars	11
Clear Text	11
Close Commands	12
Close Book	12
Close Document	12
Close TextFile	13
Copy Text	13
Cut Text	14
DialogBox commands	14
DialogBox--ChooseFile	15
DialogBox--Prompt for Data	16
DialogBox--ScrollBar	17
DialogBox--Multi-Edit	18

Delete Commands	22
Delete CondTag	22
Delete File	23
Delete Object	23
Delete TableColumns	24
Delete TableRows	25
Delete Text	25
Delete TextInsetContents	26
Delete Var	26
Demote	27
Display	28
DocCompare	28
Event EventName	29
Execute Commands	30
Exec Compile Command	30
Execute FrameCommand (Fc)	31
Execute Hypertext	32
Execute StartUndoCheckPoint	32
Execute EndUndoCheckPoint	33
Execute ClearUndoHistory	33
Exec Script Command	34
Exec Wait Command	34
Export Command	34
Export Object	34
Find Commands	37
Find Member	37
Find String (In String)	38
Find String (In Frame Object)	39
Find TextRange	40
Frame Find from text location	40
Function Command	43
Generate BookFiles	43
Get Commands	44
Get BookConditionalBuildExpression(<i>FM 2015 or greater</i>)	44
-----	44
Get BookConditionalSettings(<i>FM 2015 or greater</i>)	44
-----	45
Get Member	45

Get Object - - - - -	46
Get String - - - - -	49
Get TextList - - - - -	51
Get TextProperties - - - - -	54
GlobalVar command - - - - -	55
If, Else, ElseIf, EndIf - - - - -	55
Import Commands - - - - -	56
Import File - - - - -	56
Import Formats - - - - -	62
Import ElementDefs - - - - -	63
Install Script - - - - -	64
Leave - - - - -	66
LeaveLoop - - - - -	66
LeaveScript - - - - -	67
LeaveSub - - - - -	67
Local Command - - - - -	68
Loop, EndLoop - - - - -	69
LoopNext - - - - -	73
Merge - - - - -	74
MsgBox - - - - -	74
New Commands - - - - -	75
New Anchored Objects - - - - -	79
New Anchored Formatted Objects - - - - -	80
New Format Rule Objects - - - - -	81
New FormatChangeList - - - - -	81
New FormatRule - - - - -	81
New FmtRuleClause - - - - -	82
New Graphic Objects - - - - -	83
New Menu Commands - - - - -	84
New Command - - - - -	84
New Menu - - - - -	85
New MenuItemSeparator - - - - -	86
New Named Objects - - - - -	87
New Series Objects - - - - -	88
New BodyPage - - - - -	88
New BookComponent - - - - -	88
New Pgf - - - - -	89
New Structured Objects - - - - -	90
New ElementDef - - - - -	90
New Element - - - - -	91

New Attribute	92
New AttributeDef	93
New AttributeList	94
New AttributeDefList	95
New ElementLoc	96
New ElementRange	96
New Table Objects	97
New Table	97
New TableRows	98
New TableCols	98
New Text Operations	99
New Text	99
New Simple Data Types	100
New Integer	100
New Real	101
New Metric	101
New String	101
New Object	102
New Structures	103
New TextLoc	103
New TextRange	103
New DataRecord	104
New Simple List Data	104
New IntList	105
New UIntList	105
New MetricList	105
New StringList	105
New DataList	105
New PointList	106
New TabList	106
New PropertyList	107
New LibVar	108
New ScriptVar	109
New SubVar	109
New TextFile	111
Open Commands	112
Open Project (See EsLObject reference FM 2017 or greater)	112
Open Document or Book	112
Open TextFile	119
Paste Text	120
PopClipboard	121
Print Commands	121

Print Document or Book - - - - -	121
Promote - - - - -	122
PushClipboard - - - - -	123
Quit Session - - - - -	123
Read command - - - - -	123
Remove Commands- - - - -	124
Remove Attribute - - - - -	124
Remove CommandObject - - - - -	125
Remove Member - - - - -	126
Remove MenuObject- - - - -	126
Remove MenuSepObject - - - - -	127
Remove Property - - - - -	128
Remove UndefAttr - - - - -	128
Replace Member - - - - -	129
Return Command - - - - -	130
Run Command - - - - -	131
Save Commands - - - - -	132
Save Document or Book - - - - -	132
Select Commands - - - - -	140
Select TableCells - - - - -	140
Set command - - - - -	141
Sort command- - - - -	141
Split - - - - -	142
Straddle TableCells Command- - - - -	143
Sub Command - - - - -	143
Uninstall Script - - - - -	144
Unwrap - - - - -	144
Update commands - - - - -	145
Update objTypes- - - - -	145
Update Book - - - - -	147
Wrap - - - - -	149
Write Command - - - - -	150

2 FrameMaker Object Reference - - - - - 153

Attribute Conditional Expression - - - - -	153
Object Name - AttrCondExpr - - - - -	153
Books - - - - -	153

Object name - Book	154
Object name - BookComponent	160
Character formats	165
Object Name - CharFmt	165
Colors	167
Object Name - Color	167
Combined font definitions	168
Object name - CombinedFontDefn	169
Condition Formats	169
Object name - CondFmt	169
Cross-references	170
Object name - XRef	170
Object name - XRefFmt	171
Documents	171
Object name - Doc	171
XMP Metadata	194
HypertextParseErr Table	195
HypertextValidateErr Table	195
Hypertext Command Code Table	196
Link Destination Object Type Table	198
Flows	199
Object name - Flow	199
Footnotes	200
Object name - Fn	200
Graphic Objects	200
Object name - AFrame	201
Object name - Arc	204
Object name - Ellipse	206
Object name - GraphicsFmt (FM 11 or greater)	208
-----	213
Object name - Group	213
Object name - Inset	215
Object name - Line	218
Object name - Math	219
Object name - MathML (FM 12 or greater)	222
Object name - Polygon	224
Object name - Polyline	226
Object name - Rectangle	228

Object name - RoundedRectangle	230
Object name - TextFrame	232
Object name - TextLine	235
Object name - UnanchoredFrame	237
Inline Components (FM 2015 or greater)	240
Object name - InlineComponent	240
Markers	240
Object name - Marker	240
Marker types	241
Object name - MarkerType	241
Menus, Commands, and Menu Item Separators	242
Object name - Menu	242
Object name - Command	242
Object name - MenuItemSeparator	245
Pages	246
Object name - BodyPage	246
Object name - HiddenPage	247
Object name - MasterPage	247
Object name - RefPage	248
Paragraphs	248
Object name - Pgf	248
Object name - PgfFmt	253
Rubi composites	257
Object name - Rubi	257
Session	258
Object name - Session	258
KeyCatalog	264
Object name - KeyCatalog (FM 11 or greater)	264
Structured Objects	264
Object name - Element	264
Object name - ElementDef	270
Object name - FmtChangeList	272
Object name - FmtRule	273
Object name - FmtRuleClause	274
SubColumns	275
Object name - SubCol	275
Table Objects	276
Object name - Cell	276

Object name - Column (FM 2015 or greater)	278
-----	279
Object name - Row	279
Object name - Tbl	280
Object name - TblFmt	284
Table ruling formats	286
Object name - RulingFmt	286
Text Insets	287
Object name - TiApiClient	287
Object name - TiFlow	288
Object name - TiText properties	289
Object name - TiTextTable properties	290
Text properties	291
Variables	293
Object name - Var	293
Object name - VarFmt	293

3 FrameMaker Event Names - - - - - 295

FrameMaker Events	295
FrameMaker Event parameters	295
Hypertext Event	301
FrameMaker Message parameters	301
ElmScript Events	301
ElmScript Menu Events	302

4 F-Codes - - - - - 303

FrameMaker Codes	303
------------------	-----

Command Option Index - - - - - 333

Property Name Index - - - - - 343

Chapter 1

Command Reference

This chapter lists the ElmScript functions alphabetically. If you know the name of a function and want a complete description of it, look it up in this chapter. If you know what you want to do, but don't know which function to use see the function summary.

Command Format

Most ElmScript commands have the following general format:

```
CommandName Option1[(optionvalue1)] Option2[(optionvalue2)] ...
```

Some commands have many possible options; some have only a few (some don't have any at all). Most options, as the term indicates, are optional. Default values will be used and, in most cases, the default value will suffice. Some options are (yes, I know, it's an oxymoron) required. Some options require an option value, while, for others, the option name alone is enough. Required options are indicated. The default values for the optional options are also indicated.

Add Commands

Add CommandObject

The **Add CommandObject** command adds an existing user menu command (or menu separator) or a FrameMaker menu command (or menu separator) to an existing menu. FrameMaker adds the command to the bottom of the menu. Set the command object's `PrevMenuItemInMenu` and `NextMenuItemInMenu` properties, to change a its position on a menu.

Format:

```
Add CommandObject (commandobjectvarname) To(menuobjectvarname)
```

Table 1: Add CommandObject Options

Option Name	Option Description
CommandObject (<i>Required</i>)	The variable name of the command object (or menu separator object)
To (<i>Required</i>)	The variable name of the menu object.

The `commandobjectvarname` is created by the **New Command** command. The `menuobjectvarname` is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu.

Example:

The following script creates a command with the label 'User Command' and adds it to the File menu, then moves it the first position in the menu. The **New Command** command can add the command directly to the menu bar using the **AddTo** option, so you only need the **Add CommandObject** command when you wish to add a command to more than one menu.

```

. . .
Get Object Type(Menu) Name('FileMenu') NewVar(filemenuobj);
New Command Label('User Command') EventProc(UserCmdEvent)
    NewVar(cmdobj);
Add CommandObject(cmdobj) To(filemenuobj);
Set filemenuobj.FirstMenuItemInMenu = cmdobj;

. . .

Event UserCmdEvent
    MsgBox 'User Pressed the User Command Menu';
EndEvent

. . .

```

See also

“New Menu Commands” on page 84, “New Menu” on page 85, and “Get Object” on page 46

Add Member

The **Add Member** command adds an individual member to a list data item.

Format

```

Add Member(membervalue) To(listvar)
    [After(memberNum)] [Before(memberNum)];

```

Table 2: Add Member Options

Option Name	Option Description
Member	The value to add to the list. Should be the same type as the other list members
To (Required)	Specifies the list variable name.

Example

The following code adds the new member to the string list:

```

. . .
New StringList NewVar(gvFruitList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Add Member('Grapes') To(gvFruitList);
. . .

```

The list will now contain four values 'Apples', 'Oranges', 'Peaches', and 'Grapes'.

Add MenuObject

The **Add MenuObject** command adds an existing user menu or a FrameMaker menu to an existing menu or menu bar. FrameMaker will add the menu to the bottom of the menu. Set the menu object's `PrevMenuItemInMenu` and `NextMenuItemInMenu` properties, to change it's position on a menu.

Format:

```
Add MenuObject(menuobjectvarname1) To(menuobjectvarname2)
```

Table 3: Add MenuObject Options

Option Name	Option Description
MenuObject (Required)	The variable name of the menu object.
To (Required)	The variable name of the menu or menu bar object

The `menuobjectvarname1` is created by the **New Menu** command. The `menuobjectvarname2` is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu.

Example:

The following script creates a menu labeled 'My New Menu' and adds it the main FrameMaker menu bar. It also creates a command with the label 'User Command' and adds it to the newly created menu. The **New Menu** command can add the menu directly to the menu bar using the `AddTo` option, so you only need the **Add MenuObject** command when you wish to add a menu to more than one menu or menubar.

```
. . .
New Menu Label('My New Menu') NewVar(newmenuobj)
Add MenuObject(newmenuobj) To('!MakerMainMenu');
New Command Label('User Command') EventProc(UserCmdEvent)
    NewVar(newmenuobj);

. . .

Event UserCmdEvent
    MsgBox 'User Pressed the User Command Menu';
EndEvent

. . .
```

See also

"New Menu" on page 85, "Remove MenuObject" on page 126, and "Get Object" on page 46.

Add MenuSepObject

The **Add MenuSepObject** command adds an existing menu separator or a FrameMaker separator to an existing menu.

FrameMaker adds the separator to the bottom of the menu. Set the separator object's `PrevMenuItemInMenu` and `NextMenuItemInMenu` properties, to change a it's position on a menu.

Format:

```
Add MenuSepObject(sepobjectvarname) To(menuobjectvarname)
```

Table 4: Add MenuSepObject Options

Option Name	Option Description
MenuSepObject (<i>Required</i>)	The variable name of the separator object.
To (<i>Required</i>)	The variable name of the menu object.

The `sepobjectvarname` is created by the **New MenuSeparator** command or retrieved by the **Get Object** command. The `menuobjectvarname` is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu.

Example:

The following script creates a command with the label 'User Command' and adds it to the newly created 'My New Menu' menu, then adds a menu item separator.

```

. . .
New Menu Label('My New Menu') NewVar(newmenuobj)
    AddTo('!MakerMainMenu');
New Command Label('User Command') EventProc(UserCmdEvent)
    NewVar(newmenuobj) AddTo(newmenuobj);
Get Object Type(MenuItemSeparator) Name('Separator1')
    NewVar(sepobj);
Add MenuSepObject(sepobj) To(newmenuobj);

. . .
Event UserCmdEvent
    MsgBox 'User Pressed the User Command Menu';
EndEvent

. . .

```

See also

“New MenuItemSeparator” on page 86, “Remove MenuSepObject” on page 127, and “Get Object” on page 46.

Add Property

The **Add Property** command adds an individual property to a property list item.

Format

```
Add Property To(propolistvar) PropertyName(PropertyValue);
```

Table 5: Add Property Options

Option Name	Option Description
To (<i>Required</i>)	Specifies the property list variable name.
PropertyName	The name of the property. This can be any property for any object.
PropertyValue	A value that is appropriate for the property.

Example

The following code creates a PropertyList and adds the PgfAlignment property to property list with a value of PgfLeft:

```

. . .
New PropertyList NewVar(nprops) Name('CellBody') FontSize(12) KeepWithPrev(True);

Add Property To(nprops) PgfAlignment(PgfLeft);
. . .

```

The list will now contain four properties.

See also

“New PropertyList” on page 107

Apply Commands

Apply AttrExpression

(FM 9.0 or greater)

The **Apply AttrExpression** command applies an attribute expression to a document to perform attribute-based filtering. This command can also (optionally) apply an attribute-based expression to the document where the filtered text is converted to conditional text, (FM 10 or greater).

Format:

```

Apply AttrExpression AttrExpr(attrExprObject)
  [CondFmt(condFmtObject)] [Remove(True/False)]; (FM 10 or greater)

```

Table 6: Apply AttrExpression Options

Option Name	Option Description
AttrExpr (<i>Required</i>)	The Attribute expression object (AttrCondExpr) to apply.
CondFmt	The condition format (CondFmt) object for the conditional text option (optional) <i>FM 10 or greater</i>
Remove	True to remove the conditional text settings at all locations in the document where the condFmtObject has been applied (optional, requires the CondFmt option) <i>FM 10 or greater</i>

Example 1

The following script applies the specified attribute cond expression to the current document:

```

. . .
Get Object Type(AttrCondExpr) Name('MyAttrCondExpr') NewVar(gvAttrCondObj);
Apply AttrExpression AttrExpr(gvAttrCondObj);
. . .

```

Example 2

The following script applies the specified attribute cond expression to the current document applying the `:MyCondFmt` condition format to the filtered text, removing the conditional text settings in the document.

```

. . .
Get Object Type(AttrCondExpr) Name('MyAttrCondExpr') NewVar(gvAttrCondObj);
Get Object Type(CondFmt) Name('MyCondFmt') NewVar(gvCondFmtObject);
Apply AttrExpression AttrExpr(gvAttrCondObj) CondFmt(gvCondFmtObject) Remove(True);
. . .

```

See also

“New Named Objects” on page 87 and “Get Object” on page 46.

Apply BookConditionalBuildExpressions

(FM 2015 or greater)

The **Apply BookConditionalBuildExpressions** command is used to add, edit or delete conditional expressions to the current specified book.

Format:

```
Apply BookConditionalBuildExpressions [BookObject(bookObj)]
    [AddEditTags(addlist)] [DeleteTags(dellist)];
```

Table 7: Apply BookConditionalBuildExpressions Options

Option Name	Option Description
BookObject	The book object variable. If not specified, the currently active book will be used.
AddEditTags	A stringlist data item containing a pair of values, the first of which is the BuildExpression name and the second it the build expression value. You can have as many of these pairs as you like.
DeleteTags	A stringlist data item containing a list of Build expression names that you wish to delete.

Example

The following script adds two build expressions and deletes one build expression from the currently active book. This will set the condition mode to ShowAsPerExpression and sets the build expr to 'MyBuildExpr' .

```
. . .
New StringList NewVar(gvAdd) 'MyBuildExpr' '"Comment"OR"MyCondTag'
    'MyBuildExpr2' '"Comment"OR"MyOtherCondTag' ;
New StringList NewVar(gvDelete) 'MyBadBuildExpr';
Apply BookConditionalBuildExpressions AddEditTags(gvAdd) DeleteTags(gvDelete);
If ErrorCode=0
    Display 'No Error';
Else
    Display 'Error-'+ErrorMsg;
EndIf
. . .
```

Apply BookConditionalSettings

(FM 2015 or greater)

The **Apply BookConditionalSettings** command applies conditional settings in the selected book based on the specified settings.

Format:

```
Apply BookConditionalSettings [BookObject(bookObj)] Settings(dataRecordObject);
```


Table 8: Apply BookConditionalSettings Options

Option Name	Option Description
BookObject	The book object variable. If not specified, the currently active book will be used.
Settings (Required)	<p>The setting values for this command are fields in a DataRecord data type. You first create a DataRecord data item (New DataRecord), then use the fields in the data record as settings to apply to the specified book.</p> <p>The values in the data record are as follows:</p> <p>ShowState (ShowAll, ShowAsPerConditions, ShowAsPerExpression)</p> <p>ShowConditions-A StringList of condition names in the show state</p> <p>HideConditions-A StringList of condition names in the hide state</p> <p>ActiveConditionalExpression-String value with the active condition expression tag</p> <p>ShowConditionIndicators-Show/Hide condition indicators</p> <p>ApplyConditionalSettingsToViewOnlyDoc-True, Apply the conditions to view only documents</p> <p>ApplyConditionalSettingsToNestedBooks-True, Apply the conditions to nested books</p> <p>ApplyConditionalSettingsShowBookErrorLog-True, Show errors in the book error log.</p>

Example

The following script applies the specified settings to the currently active book. This will set the condition mode to ShowAsPerExpression and sets the build expr to 'MyBuildExpr'.

. . .

```

New DataRecord NewVar(gvSettings) ShowState(ShowAsPerExpression)
  ActiveConditionalExpression('MyBuildExpr');
Apply BookConditionalSettings Settings(gvSettings);
If ErrorCode=0
  Display 'No Error';
Else
  Display 'Error-'+ErrorMsg;
EndIf

```

. . .

Apply PageLayout

The **Apply PageLayout** command applies the source page's layout to the destination page.

Format:

```

Apply PageLayout Sourcepage(pagevarname)
  Destpage(destpagevarname);

```

Table 9: Apply PageLayout Options

Option Name	Option Description
Destpage	The Destination page object variable. If not specified, the current active document's current page is used.
Sourcepage (Required)	The Source page object variable

Example

The following script applies the layout of the Left master page to the current page:

```

. . .
Apply PageLayout Sourcepage(LeftMasterPage);
. . .

```

See also

“Import Formats” on page 62.

Apply TextProperties

The **Apply TextProperties** command sets the text properties (such as the format tag, font family, and size) for a text range. It can also apply conditional formats to a range of text.

Format:

```

Apply TextProperties
  DocObject(docobjectvar) or TextRange(textrangevar)
  Properties(propertiesvar)
and/or
  CondFmt(conditionformatname) [CondFmt('condname') ... ]

```

Table 10: Apply TextProperties Options

Option Name	Option Description
DocObject	The document object variable. If this option is chosen, then the properties will be apply to the current text selection of this document.
TextRange	The TextRange variable. If this option is chosen, then the properties are applied to this text selection.
Properties	The property list to apply to the text range. If there is neither a docobjectvar nor a textrangevar specified, then the properties are applied to the current text selection of the current active document. If there is no current active document, the command fails.
CondFmt	The name of a conditional format. The condition format must already exist in the document. You may repeat this option as many times as necessary to define a set of conditions for a range of text.

Example 1:

The following script applies the text properties of the first paragraph in the document to the current text selection:

```

. . .
Set pgfobj = FirstPgInDoc;
Apply TextProperties
  Properties(pgfobj.properties);
. . .

```

Example 2:

The following script applies the condition formats 'Hidden' and 'FirstTime' to the current text selection:

```

. . .
Apply TextProperties CondFmt('Hidden') CondFmt('FirstTime');
. . .

```

See also

- “Get TextProperties” on page 54.
- “New ConditionFormat” on page 87.

CallClient

The CallClient command allows you to send a message to another Frame Api Client or another script.

Format 1:

```

CALLCLIENT FrameClient (clientname) Message(msg) [ReturnVal(retVar)]
[ReplaceWithNull(char)];

```

Format 2:

```

CALLCLIENT ScriptName (scriptname) Message(msg) [ReturnVal(retVar)]
[ReplaceWithNull(char)];

```

Table 11: CallClient Options

Option Name	Option Description
FrameClient	<p>The name of the external Frame Api client. This name is the name client as defined in the Frame ini file for windows and the filename for the Macintosh. For windows: clientName=Standard, Some text, c:\ApiClientDir\apiclient.dll</p> <p>Here clientName is the name of the client. It is up to the actual Api client to accept this message and do something with it. Certain clients are built to respond this way.</p>
ScriptName	<p>The name of a ElmScript event script that is currently installed. This script must identify the NoteCallClient event. This event is called for the identified script.</p>
Message	<p>The text of the message to send.</p>
ReplaceWithNull	<p>A character (1 character string) used as a substitute for a null byte. Some external programs require that the message is a list of strings each separated by a NULL (zero) byte and terminated by 2 null bytes. ElmScript does not support this kind of data, so you built a standard string and use a chosen character to act as a substitute for the null byte. Specify that character here and ElmScript will convert it to a null byte before sending it to the specified client. The primary use for this currently is the WebDAV client, which requires data in this form.</p>
ReturnVal	<p>The value returned from the client, if any.</p>

Example 1

The following code sends a message to a previously installed event script. In this example, that script simply displays the message passed:

```

. . .
CallClient ScriptName('CALLTEST')
    Message ('Test Message to Client')
    ReturnVal(rval);

If rval < 0
    SET errorcode = rval;
EndIf
Display 'Return-'+rval+' Msg-'+ErrorMsg;+

--->>> this is in a separately installed script by the name of CALLTEST
Event NoteClientCall

    Display 'Doc='+FrameDoc+' Msg-'+FILENAME;

    Return Value(42);

EndEvent

```

Example 2

The following code sends a message to a third party client called SetPrint which sets the default FrameMaker printer to the specified printer, in this case the Acrobat Distiller: You must have the SetPrint API client installed for this to work.

```

. . .
CallClient FrameClient('SetPrint') Message('/save') ReturnVal(vRet);
CallClient FrameClient('SetPrint') Message('Acrobat Distiller') ReturnVal(vRet);

...

CallClient FrameClient('SetPrint') Message('/restore') ReturnVal(vRet);

```

Example 3

The following code sends a message to WebDAV client. The command is a set of strings separated by null bytes. The '#' character substitutes for the null byte while the string is being built. The ReplaceWithNull option changes all the '#' characters to null bytes before sending the string to WebDAV. This is only an example. The values here are not valid. You must get the correct values for your own WebDAV server.

```

. . .
Set vDAVcmd = 'SetServer' + '#' +
    'FTest' + '#' +
    'http://SomeId.ServeItToMe.com:12300/TestFolder/' + '#' +
    'AAA' + '#' +
    'AAA' + '#' +
    'C:\WorkGroupTest' + '##' +
    '####'; // Add some extra null bytes at the end

CallClient FrameClient('WebDAV') Message(vDAVcmd)
    ReplaceWithNull('#') ReturnVal(retVar);

...

```

Clear

Clear ChangeBars

The **Clear ChangeBars** command clears all change bars from a specified document.

Format:

```
Clear ChangeBars [DocObject(docobjectvar)]
```

Table 12: Clear Changebars Options

Option Name	Option Description
DocObject	The object variable of the document to clear the change bars. If not specified the change bars will be cleared from the current active document

Example

The following code removes the change bars from the current document:

```
. . .
Clear ChangeBars;
. . .
```

Clear Text

The **Clear Text** command deletes the current selection from the specified document.

Format:

```
Clear Text [DocObject(docobjectvar)] [clearoptions]
```

Table 13: Clear Text Options

Option Name	Option Description
DocObject	The document object variable from which to clear the selection. If not specified, the current active is used.
Interactive	Clear option: Notify user for options.
CutTblCells	Remove cleared table cells
VisibleOnly	Clear only the visible portion of the selection
DontDeleteHiddenText	Don't delete hidden text

Example

The following script clears the current selection of the current active document and remove the cleared table cells:

```
. . .
Clear Text CutTblCells;
. . .
```

See also

“Copy Text” on page 13, “Cut Text” on page 14, “Paste Text” on page 120.

Close Commands

The following commands close various file types in a ElmScript script.

Close Book

The **close book** command closes a FrameMaker book. If a book has unsaved changes the command will fail unless you specify the IgnoreMods option.

Format:

```
Close Book [BookObject(bookobjectvar)] [IgnoreMods];
```

Table 14: Close Book Options

Option Name	Option Description
BookObject	The object variable of the book to close. If the book object is not specified, the currently active book is closed.
IgnoreMods	Close the document whether there are unsaved changes or not.

Example

The following code closes the active book, ignoring any unsaved changes:

```
. . .
Close Book IgnoreMods

. . .
```

See also

“Save Document or Book” on page 132 and “Open Document or Book” on page 112

Close Document

The **close document** command closes a FrameMaker document. If a document has unsaved changes the command will fail unless you specify the IgnoreMods option.

Format:

```
Close Document [DocObject(docobjectvar)] [IgnoreMods]
```

Table 15: Close Document Options

Option Name	Option Description
DocObject	The object variable of the document to close. If the docobject is not specified, the currently active document is closed.
IgnoreMods	Close the document whether there are unsaved changes or not.

Example

The following code closes the active document, ignoring any unsaved changes:

```
. . .
Close Document IgnoreMods;
. . .
```

See also

“Save Document or Book” on page 132 and “Open Document or Book” on page 112

Close TextFile

The `Close TextFile` command closes a text file.

Format:

```
Close TextFile Object(fileobjectvar);
```

Table 16: Close TextFile Options

Option Name	Option Description
<code>Object</code> (<i>Required</i>)	The object variable of their textfile to close.

Example

The following code closes the active document, ignoring any unsaved changes:

```
. . .
Close Textfile Object(gvFileVar);
. . .
```

Copy Text

The `Copy Text` command copies the current selection to the clipboard.

Format:

```
Copy Text [DocObject (docobjectvar)] [copyoptions]
```

Table 17: Copy Text Options

Option Name	Option Description
<code>DocObject</code>	The document object variable from which to clear the selection. If not specified, the current active is used.
<code>Interactive</code>	Copy option: Notify user for options.
<code>VisibleOnly</code>	Copy only the visible portion of the selection

Example

The following script copies visible text of the current selection of the current active document to the clipboard:

```

. . .
Copy Text VisibleOnly;
. . .

```

See also

“Clear Text” on page 11, “Cut Text” on page 14, and “Paste Text” on page 120.

Cut Text

The **Cut Text** command deletes the current selection from the specified document and copies the contents to the clipboard.

Format:

```
Cut Text [DocObject (docobjectvar)] [cutoptions]
```

Table 18: Cut Text Options

Option Name	Option Description
DocObject	The document object variable from which to clear the selection. If not specified, the current active is used.
Interactive	Cut option: Notify user for options.
CutTblCells	Cut option: Remove cleared table cells
VisibleOnly	Cut option: Clear only the visible portion of the selection
DontDeleteHiddenText	Cut option: Don't delete hidden text

Example

The following script cuts the current selection of the current active document to the clipboard and prompts the user if hidden text or table cells are selected:

```

. . .
Cut Text Interactive;
. . .

```

See also

“Clear Text” on page 11, “Copy Text” on page 13, and “Paste Text” on page 120.

DialogBox commands

The **DialogBox** command displays several different styles of dialog boxes. These dialog boxes give you several ways to get information from the user.

DialogBox--ChooseFile

Format:

```
DialogBox Type(ChooseFile)
  [Directory(directoryName)] [CheckDir(check)] [Title(dialogTitle)]
  [Init(initialstring)] NewVar(varname) [Button(buttonvar)]
  [Mode({SelectFile OpenFile SaveFile OpenDirectory})];
```

This dialog box allows the user to select a file name to open or save, or a directory name.

Table 19: DialogBox ChooseFile Options

Option Name	Option Description
Type (Required)	This option identifies the dialog as a ChooseFile dialog box.
Title	A string expression containing the title of the dialog box.
Directory	The initial directory to start the file selection.
CheckDir	If the directory option is specified, check if the directory exists first. The default value is True. To turn this off, set this value to False.
Init	The initial value to put in the selection text line.
NewVar	The variable name to put the file name that the user selected.
Button	The variable name to put the value of the button that the user pressed. The value returned in the variable will be one of the following: OkButton User pressed the Ok button. CancelButton User pressed the Cancel button.
Mode	This dialog box has several forms depending on the type of file selection you wish. The possible values are: SelectFile OpenFile SaveFile OpenDirectory

Example:

This sample displays a dialog box allowing the user to select a file name. The selection will start in the c:\maker directory. It will have the *.fm in the text box. After the user clicks on a button the value in the text box is copied to gvFmFileName and the button pressed is copied to gvButtonVar.

```
. . .
DialogBox Type(Choosefile) Title('Please select a file name')
  Directory('c:\TestDocs') Mode(OpenFile) Button(gvButtonVar)
  NewVar(gvFmFileName) Init('*.fm');
if gvButtonVar = OKButton
  MsgBox 'The user pressed the OK button and selected this filename-'+gvFmFileName;
EndIf
```

DialogBox--Prompt for Data

Format:

```
DialogBox Type({Int Metric String}) [Title(titlestring)]
  [Init(initialvalue)] [Units({Inch mm cm pica didot})]
  [NewVar(varname)] [Button(buttonvarname)];
```

This format allows the user to enter a single value of predefined type.

Table 20: DialogBox Enter Simple Data Options

Option Name	Option Description
Type (Required)	This option identifies the dialog as a simple data input type. The allowable values are: Int - Input an integer. Metric - Input a metric value. String - Input a string value.
Title	A string expression containing the title of the dialog box.
Init	The initial value to put in the text box.
Units	If the type value is metric, you may specify the default units of data the user will enter if there are no units specified on the input line. You may specify one of the following values: INCH POINTS CM MM PICA DIDOT CICERO
NewVar	The variable name to put the data that the user entered.
Button	The variable name to put the value of the button that the user pressed. The value returned in the variable will be one of the following: OkButton User pressed the Ok button. CancelButton User pressed the Cancel button.

Example:

This example displays a dialog box prompting the user to enter a metric value.

```
. . .
DialogBox Type(Metric) Title('Enter a Metric value for the size of something')
  Init(3) Units(Inch) NewVar(gvMetricVar) Button(gvButtonVar);
if gvButtonVar = OKButton
  MsgBox 'The user pressed the OK button and entered this data-'+gvMetricVar;
EndIf
. . .
```

DialogBox--ScrollBox

Format 3:

```
DialogBox Type(ScrollBox) [Title(titlestring)]
  [Caption(titlebarText)]
  [Init(initialselection)] List(listvar)
  [NewVar(varname)] [Button(buttonvarname)]
  [ReturnIndex(idxvarname)];
```

Table 21: DialogBox Scrollbox Options

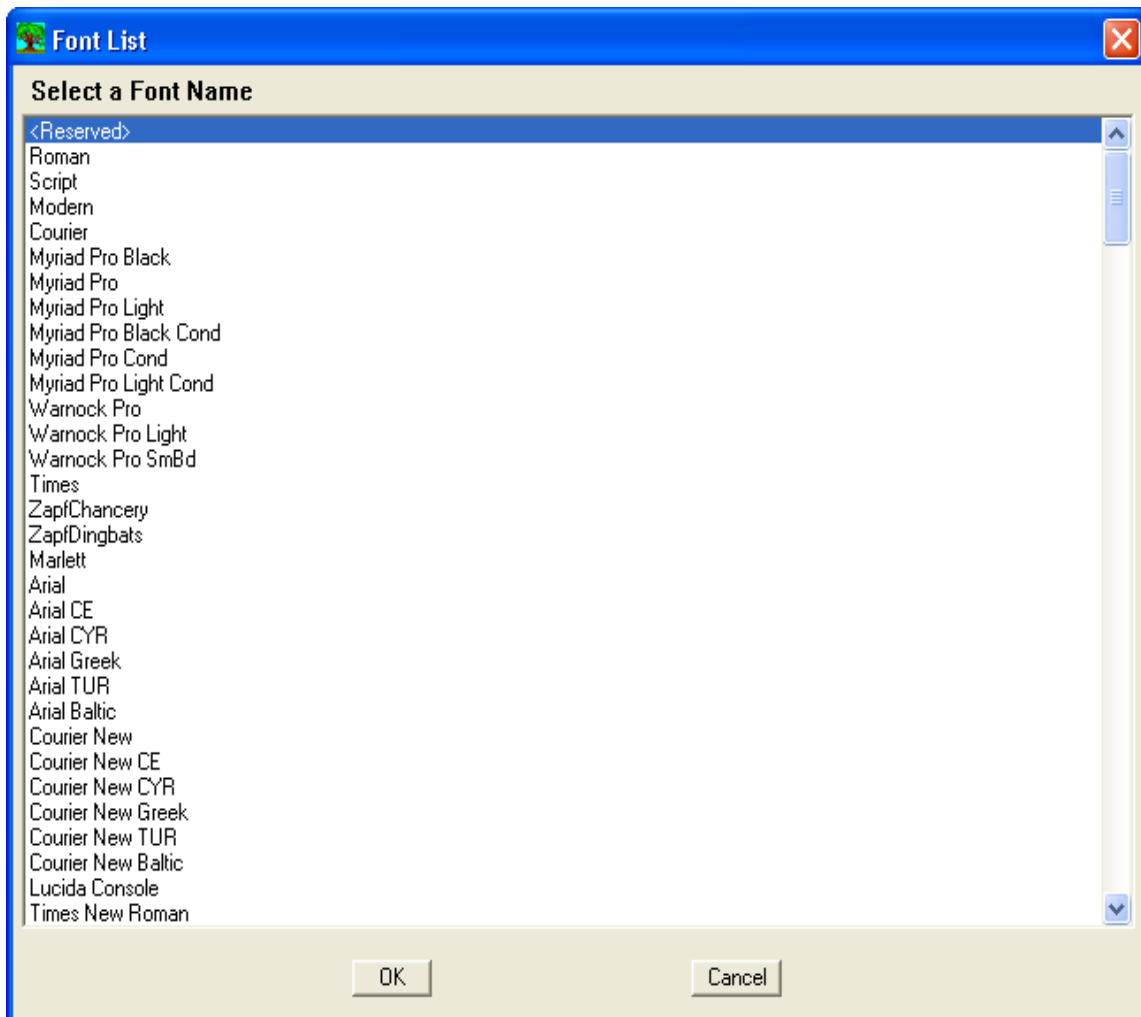
Option Name	Option Description
Type (Required)	This option identifies the dialog as a scrollbar. The value is ScrollBox.
Caption	A string expression containing the caption of the dialog box in the title bar.
Title	A string expression containing the title of the dialog box.
Init	The initial number of the selected member. If this value is -1, then no item is selected.
List	This is a stringlist variable containing the list of strings to display in the scrolling dialog box.
NewVar	The variable name to put the string selected.
Button	The variable name to put the value of the button that the user pressed. The value returned in the variable will be one of the following: OkButton User pressed the Ok button. CancelButton User pressed the Cancel button.
ReturnIndex	The variable name to put the index of the string selected (starts at 1).

Example:

This example displays the list of fonts (from the Session property FontFamilyNames) and allows the user to select one of them, see “ScrollBox DialogBox” on page 18.

```
DialogBox Type(ScrollBox) Title('Select a Font Name')
  Caption('Font List')
  Init(-1) List(FontFamilyNames)
  NewVar(gvFontNameVar) Button(gvButtonVar);
if gvButtonVar = OKButton
  MsgBox 'The user pressed the OK button and selected this font name-'+gvFontNameVar;
EndIf
. . .
```

Figure 1-1 ScrollBox DialogBox



DialogBox--Multi-Edit

Format:

```
DialogBox Type(MEdit) [Title(DialogTitleString)] [Title2(TitleString)]
  [Label1(Label1String)] [Label2(Label2String)] [Label3(Label3String)]
  [Edit1(Edit1Var)] [Edit2(Edit2Var)] [Edit3(Edit3Var)]
  [CheckBox1Label(Cbx1LabelString) CheckBox1(Cbx1Var)]
  [CheckBox2Label(Cbx2LabelString) CheckBox2(Cbx2Var)]
  [CheckBox3Label(Cbx3LabelString) CheckBox3(Cbx3Var)]
  [CheckBox4Label(Cbx4LabelString) CheckBox4(Cbx4Var)]
  [Button3Label(string)] [Button4Label(string)]
  [Button(buttonvarname)];
```

This format allows you to display a dialog box with up to three edit fields, up to four checkboxes and two optional command buttons. NOTE: Most of these items are optional. They will appear on the dialog box only if you specify them.

This was designed for a quick way to have the user enter a few bits of information. Use the custom forms facility to create dialog boxes that more precisely suit your needs.

Table 22: DialogBox MEdit Options

Option Name	Option Description
Type (Required)	This option identifies the dialog as a scrollbar. The value is <code>ScrollBar</code> .
Title	A string expression containing the title of the dialog box.
Title2	A string expression containing a title line inside the dialog box.
Label1	Specifies the label for the first edit field.
Edit1	Specifies the variable in which to place the data in the first edit field.
Label2	Specifies the label for the second edit field.
Edit2	Specifies the variable in which to place the data in the second edit field.
Label3	Specifies the label for the third edit field.
Edit3	Specifies the variable in which to place the data in the third edit field.
CheckBox1Label	Specifies the label for the first checkbox.
CheckBox2Label	Specifies the label for the second checkbox.
CheckBox3Label	Specifies the label for the third checkbox.
CheckBox4Label	Specifies the label for the fourth checkbox.
CheckBox1	Specifies the variable in which to place the state of first checkbox. A value of 0 means the box is unchecked and a value of 1 means the box was checked.
CheckBox2	Specifies the variable in which to place the state of second checkbox. A value of 0 means the box is unchecked and a value of 1 means the box was checked.
CheckBox3	Specifies the variable in which to place the state of third checkbox. A value of 0 means the box is unchecked and a value of 1 means the box was checked.
CheckBox4	Specifies the variable in which to place the state of fourth checkbox. A value of 0 means the box is unchecked and a value of 1 means the box was checked.
Button3Label	Specifies the label for the first optional button (the third button altogether).
Button4Label	Specifies the label for the second optional button (the fourth button altogether).
Button	<p>The variable name to put the value of the button that the user pressed.</p> <p>The value returned in the variable will be one of the following:</p> <p>OkButton User pressed the Ok button.</p> <p>CancelButton User pressed the Cancel button.</p> <p>BUTTON3 User pressed the first optional button.</p> <p>BUTTON4 User pressed the second optional button.</p>

Example:

This example display a set of data entry fields for the user to fill in.

```
DialogBox Type(MEdit) Title('Demographic Information')
  Title2('Enter Name and Address')
  Label1('Name') Edit1(gvNameVar)
  Label2('Address') Edit2(gvAddressVar)
  Label3('City,State,Zip') Edit3(gvCityStateVar)
  Button3Label('Use Default') Button(gvButtonVar);
If gvButtonVar = OKButton
  MsgBox 'The user pressed the OK button';
Else
  If gvButtonVar = Button3
    MsgBox 'User pressed the Use Default button';
  EndIf
EndIf
. . .
```

Example:

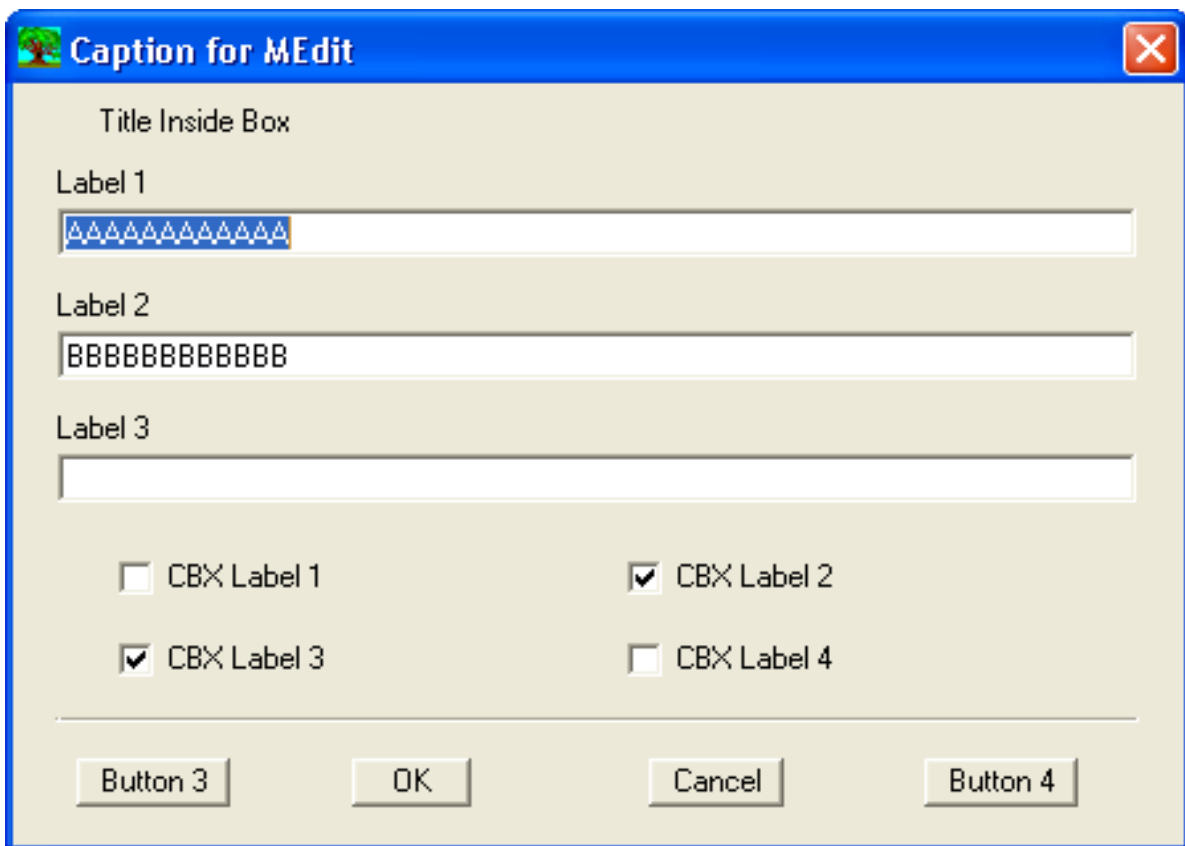
This example displays all three edit boxes along with all four check boxes, see “Multi-Edit DialogBox” on page 22.

```

SET gvEditVar1 = 'AAAAAAAAAAAA';
SET gvEditVar2 = 'BBBBBBBBBBBB';
SET gvEditVar3 = '';
SET gvCbxBVar1 = False;
SET gvCbxBVar2 = True;
SET gvCbxBVar3 = True;
SET gvCbxBVar4 = False;
DIALOGBOX Type(MEdit)
    Title('Caption for MEdit')
    Title2('Title Inside Box')
    Label1('Label 1')
    Label2('Label 2')
    Label3('Label 3')
    Edit1(gvEditVar1)
    Edit2(gvEditVar2)
    Edit3(gvEditVar3)
    CheckBox1Label('CBX Label 1')
    CheckBox2Label('CBX Label 2')
    CheckBox3Label('CBX Label 3')
    CheckBox4Label('CBX Label 4')
    CheckBox1(gvCbxBVar1)
    CheckBox2(gvCbxBVar2)
    CheckBox3(gvCbxBVar3)
    CheckBox4(gvCbxBVar4)
    Button3Label('Button 3')
    Button4Label('Button 4')
    Button(gvButtonVar)
    ;
IF gvButtonVar = OKBUTTON
    Write Console 'OK Button Pressed';
Else
    If gvButtonVar = CANCELBUTTON
        Write Console 'Cancel Button Pressed';
    Else
        If gvButtonVar = BUTTON3
            Write Console 'BUTTON3 Button Pressed';
        Else
            If gvButtonVar = BUTTON4
                Write Console 'BUTTON4 Button Pressed';
            Else
                Write Console 'Unknown Button Pressed';
            EndIf
        EndIf
    EndIf
EndIf
EndIf
. . .

```

Figure 1-2 Multi-Edit DialogBox



Delete Commands

Delete CondTag

(FM 11 or greater)

The `Delete CondTag` command .

Format:

```
Delete CondTag(condObjectVar) [Action(actionType)];
```


Table 23: Delete CondTag Options

Option Name	Option Description
CondTag (Required)	The Condition Object.
Action	The optional action value. It is a String value with one of the following values: 'Ask' 'UnCond' or 'Delete' The default value is 'Delete'.

Example:

The following code deletes the Condition in the variable `gvCondTagObject`.

```

. . .
Delete CondTag(gvCondTagObject) Action('Ask');
. . .

```

Delete File

The **Delete File** command deletes a file from your hard disk. This command will delete any file as long as its not protected.

Format:

```
Delete File(StringExpression);
```

Table 24: Delete File Options

Option Name	Option Description
File (Required)	The String variable, constant or expression specifying a file name.

Example:

The following code deletes the file 'c:\temp\test.fm' from your hard disk:

```

. . .
Delete File('c:\temp\test.fm');
. . .

```

Delete Object

The **Delete Object** command deletes an object from a document. When an object is deleted, all the objects contained in that object are also deleted!

Format:

```
Delete Object(objectvar);
```

Table 25: Delete Object Options

Option Name	Option Description
Object (<i>Required</i>)	The object variable representing the object to delete.

Example

The following script deletes all the markers in the current document:

```

. . .
Set gvMarkerVar = FirstMarkerInDoc; // Get first marker in doc
Loop While (gvMarkerVar)
  /* As each marker is deleted, its NextMarkerInDoc property
  ** becomes invalid, so it is necessary to get the next marker
  ** before deleting the current one.
  */
  SET gvDelMarkerVar = gvMarkerVar;
  SET gvMarkerVar = gvMarkerVar.NextMarkerInDoc;
  Delete Object(gvDelMarkerVar);
EndLoop
. . .

```

See also

“New FrameMaker Objects” on page 79.

Delete TableColumns

The Delete TableColumns command deletes columns from a table. To delete an entire table, use the Delete Object command.

Format:

```

Delete TableColumns TableObject(tableobjectvar)
  [StartCol(columnnumber)] [NumCols(numberofcolumns)]

```

Table 26: Delete TableColumns Options

Option Name	Option Description
TableObject (<i>Required</i>)	The object variable of the table containing the columns.
StartCol	The first column to delete. Columns are numbered from left to right, starting with 0. Default is the first column.
NumCols	The number of columns to delete. Default is 1.

Example:

The following script deletes the first two columns in the selected table in the active document:

```

. . .
Delete TableColumns TableObject(SelectedTbl) NumCols(2)
. . .

```

See also

“Delete Object” on page 23 and “New TableCols” on page 98.

Delete TableRows

The **Delete TableRows** command deletes rows from a table. This command does not allow you to delete more than one type of row at time. The range of rows you specify must be all body rows, all header rows, or all footer rows.

To delete an entire table, use the `Delete Object` command.

Format

```
Delete TableRows TableObject(tableobjectvar)
    [RowObject(rowobjectvar)] [NumRows(numberofrows)]
```

Table 27: Delete TableRows Options

Option Name	Option Description
TableObject (<i>Required</i>)	The Object variable of the table containing the rows
RowObject	The Object variable of the first row to delete. Default is the first row.
NumRows	The number of rows to delete, including specified row. Default is 1.

Example:

The following script deletes the first row of the selected table:

```
. . .
rowobj = SelectedTbl.FirstRowInTbl;
Delete TableRows TableObject(SelectedTbl)
    RowObject(rowobj) NumRows(1);
. . .
```

See also:

“Delete Object” on page 23 and “New TableRows” on page 98.

Delete Text

The **Delete Text** command deletes the text in a specified text range in a document.

Format

```
Delete Text TextRange(textrangevar);
```

Table 28: Delete Text Options

Option Name	Option Description
textrangevar (<i>Required</i>)	The text range to delete

Example:

The following script gets the text selection from the active document and deletes it:

```
. . .
Delete Text TextRange(TextSelection)
. . .
```

See also

“New Text” on page 99.

Delete TextInsetContents

The **Delete TextInsetContents** command deletes the text in a text inset. You must unlock a text inset before you do this command. After you are done, you must relock the text inset.

Format:

```
Delete TextInsetContents(insetobjectvar)
```

Table 29: Delete TextInsetContents

Option Name	Option Description
TextInsetContents (<i>Required</i>)	The text inset object variable.

Example:

This example deletes contents of a Text Inset.

```
set gvTiVar.TiLocked = False;
Delete TextInsetContents (gvTiVar);
set gvTiVar.TiLocked = True;
```

Delete Var

The **Delete Var** command deletes a variable from your ElmScript script data space. Ordinarily you wouldn't need to delete variables; they go away when the script terminates. Occasionally, though, when variables use a large amount of memory, such as large text fields or string lists, you should delete them when you don't need them anymore. This is especially true of Event Scripts, where the variables stay around until you delete them or the Frame Session terminates.

IMPORTANT: Deleting a variable containing an Object value does not delete the object itself. It only deletes the variable containing the ID of the object. To delete an object (such as a paragraph, anchored frame, etc.), you need to use the Delete Object command.

Format:

```
Delete Var(varname);
```

Table 30: Delete Var Options

Option Name	Option Description
Var (<i>Required</i>)	The variable to delete.

Example

The following script deletes the large string variable:

```

. . .
set gvPgftext = FirstPgftext.doc.text;
. . .
Delete Var(gvPgftext);
. . .

```

Demote

The **Demote** command demotes the selected structural element or elements from the specified document. The element becomes a child of the sibling element in front of it. *Structured FrameMaker only.*

Format:

```

Demote [DocObject (docvar)] [BookObject(bookvar)]
       [Element(eltvar)] [ElementRange(elrangeVar)];

```

Table 31: Demote Options

Option Name	Option Description
DocObject	The document object variable from which to demote the element. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to demote the element. If not specified and the Doc object is also not specified, the currently active book is used.
Element	The element object to demote. If not specified and the ElementRange is also not specified, then the current element selection will be used.
ElementRange	The range of elements to demote. If not specified and the Element is also not specified, then the current element selection will be used.

Example

The following script demotes the current element selection of the currently active document.

```

. . .
Demote;
. . .

```

Example

The following script demotes the first child of the current element.

```

. . .
SET eltvar = ElementSelection.begin.child;
SET childvar = eltvar.FirstChildElement;
Demote Element(childvar);
. . .

```

See also

“Unwrap” on page 144, “Promote” on page 122, and “Wrap” on page 149.

Display

The display command will display an expression in a message box. You may display any variable or property. If the property is a list property, it will display each member of the list in successive message boxes. The user may cancel the command at each step. This command is very useful for testing and debugging a script.

Format:

```
Display Expression Expression ... Expression
```

Table 32: Display Options

Option Name	Option Description
Expression	Any expression.

Example

The following script displays the fonts in the FrameMaker system:

```
. . .
Display FontFamilyNames;
. . .
```

DocCompare

The DocCompare command lets you perform the document compare function similar to the FrameMaker UI's File->Utilities->Compare Documents function.

Format:

```
DocCompare OldDocObject(oldDocVar) NewDocObject(newDocVar)
  [InsertCondName(insCondFmtName)] [DeleteCondName(delCondFmtName)]
  [DeleteText(delTextString)]
  [Threshold(percent)]
  [SummaryOnly] [ChangeBars] [Hyperlinks] [OpenSummary] [OpenComposite]
  [Attributes] (for structured docs)
  [ReturnComposite(rcompDocVar) [ReturnSummary(rsumDocVar)];
```

Table 33: DocCompare Options

Option Name	Option Description
OldDocObject (Required)	The document object variable that is designated as the older document.
NewDocObject (Required)	The document object variable that is designated as the newer document.
InsertCondName	The condition format name tag to apply to insertions shown in the composite document.
DeleteCondName	The condition format name tag to apply to deletions shown in the composite document.
DeleteText	The string to appear in place of the deleted text.

Table 33: DocCompare Options

Option Name	Option Description
Threshold	Use 0 to use the default value of 75. Use 1 to use the value in the ThresholdFactor value in the maker.ini file. The value means the percentage of words that can change before paragraphs are considered not equal. Default value is 75.
SummaryOnly	Generate a summary document but not a composite document.
ChangeBars	Use changebars in the composite document.
Hyperlinks	Use hyperlinks in the summary document.
OpenSummary	Open the Summary document.
OpenComposite	Open the Composite document.
Attributes	Only for Structured Docs. Compare attributes for elements. <i>FM 10 or greater</i>
ReturnSummary	Variable for the Summary document object.
ReturnComposite	Variable for the Composite document object.

Example:

The following script compares documents `c:\docs\doc1.fm` and `c:\docs\doc2.fm`.

```

. . .
Open Document File('c:\docs\doc1.fm') NewVar(docVar1);
Open Document File('c:\docs\doc2.fm') NewVar(docVar2);

DocCompare OldDocObject(docVar1) NewDocObject(docVar2)
  InsertCondName('CondName1') DeleteCondName('CondName2')
  ChangeBars HyperLinks OpenSummary OpenComposite
  ReturnSummary(sumDocVar) ReturnComposite(compDocVar);

```

See also

“Open Document or Book” on page 112.

Event EventName

The **Event EventName** command indicates the start of a new procedure that will process a FrameMaker event. Events include responses to FrameMaker Events (notifications) (such as before and after opening a document) and menu command EventProcs.

Format:

```

Event eventname
. . .
EndEvent

```

In between the Event and EndEvent lines there can be any number of commands.

Table 34: Event Options

Option Name	Option Description
eventname (<i>Required</i>)	The name of the event, either predefined as in notifications or a user defined name as in menu command procedure.
EndEvent (<i>Required</i>)	The command that terminates an event.

Example:

The following script illustrates a notification event: This one displays a message showing the file name whenever someone opens a document.

```

. . .
Event NotePreOpenDoc
    MsgBox 'The user just opened a document='+FileName;
EndEvent

```

Execute Commands

Exec Compile Command

The Exec Compile command allows you to programmatically compile scripts. You can write a script to compile all your other scripts. This is useful if you have many scripts and you need to re-compile all of them.

Format:

```

Exec Compile FileName(fileNameString) OutputFileName(outFileNameString);
or
Exec Compile ScriptText(stringValue) OutputFileName(outFileNameString);

```

Table 35: Exec Compile Options

Option Name	Option Description
FileName	A string value containing the name of the script file.
ScriptText	A string value containing the text of the script. You must use one of FileName or ScriptText
OutputFileName	The name of the file to place the compiled script.

Execute FrameCommand (Fc)

The **Execute FrameCommand** command sends what are known as f-codes to FrameMaker. *F-codes* are hexadecimal codes that specify individual user actions, such as cursor movement and text entry. FrameMaker executes each f-code as if the user performed the action. There are several ways to specify a F-codes. For the first, ElmScript provides a list of identifiers which represent these codes. See “FrameMaker Codes” on page 303 for a complete list of F-code identifiers. For the second, you can specify the integer value using the Expr option. You have to know the integer code itself to do this however. The last method is for finding the F-code for FrameMaker menu commands. Most menu commands have an associated F-code. If you use the MenuCommand option with the name of the menu command supplied as a string, then it will find the F-code for that menu command. This is especially useful when changing FrameMaker versions. A menu command F-code in one version of FrameMaker will not necessarily be the same one in the next version.

You can use as many F-codes on one command as you wish. You can even mix the 3 types.

This command uses the current focus in a visible document. If you want to execute a set of f-codes in a particular document, make sure that the document is active (or has focus).

Format

```
Execute FrameCommand [fcIdentifier] [fcIdentifier] ...
                    [Expr(integerValue)] ...
                    [MenuCommand('MenuName')] ...
```

Table 36: Execute FrameCommand Options

Option Name	Option Description
fcIdentifier	An F-code identifier.
Expr	An expression that evaluates to an Integer specifying the F-code value. You can use the FCode utility function from the eUtl object to get the integer value of a F-code identifier. E.g. <code>Set gvFcode = eUtl.FCode{'HighPgf'};</code>
MenuCommand	The name of the FrameMaker menu command. This is not the Label that appears on the screen, but the actual menu name. This information is available in the FrameMaker config files, located in the fminit folder. The actual location is platform dependent.

Example:

This example toggles the Snap checkmark on the Graphic menu. The same as if the user clicked the 'Snap' menu item.

```
Execute Fc KbdSnap;
```

Example:

This example runs the NewDocument FrameMaker menu command which is File->New->Document... on the FrameMaker user interface.

```
Execute Fc MenuCommand('NewDocument');
```

Example:

This example gets the integer value of the HighPgf F-code identifier and uses it in the expression.

```
Execute Fc Expr(eUtl.FCode{'HighPgf'});
```

Execute Hypertext

The Execute Hypertext command simulates a user-invoked hypertext command. Doing this command has the same effect as a user clicking on a hypertext button.

Format:

```
Execute HyperText [DocObject(docobjectvar)]
                Command(commandstring);
```

Table 37: Execute Hypertext Options

Option Name	Option Description
DocObject	The document object variable specifying the document. If not specified, the current active document is used.
Command (Required)	The text of the hypertext command just as you would type into the hypertext marker from the user interface.

For a complete list of hypertext commands, see the FrameMaker product documentation.

Example 1:

This example executes the 'gotolink' hypertext command on the currently active document.

```
Execute Hypertext Command('gotolink xyz');
```

Example 2:

This example executes the a system command. This one starts the window notepad program with the text file, test.txt.

```
Execute Hypertext Command('message system notepad.exe test.txt');
```

Execute StartUndoCheckPoint

The Execute StartUndoCheckPoint command records the starting point of the series of commands. This allows you to treat a series of commands as one undo operation. The string in the description option will appear in the undo and redo menus and also in the history palette. A new Exec StartUndoCheckPoint command after another Exec StartUndoCheckPoint command (nested case) will have no effect and will continue grouping the commands. Requires *Frame 7.2 or greater*. See the Users Guide for more information on the multiple undo/redo features.

Format:

```
Execute StartUndoCheckPoint [DocObject(docobjectvar)]
                Description(descString);
```

Table 38: Execute StartUndoCheckPoint Options

Option Name	Option Description
DocObject	The document object variable specifying the document. If not specified, the current active document is used.
Description	The text of the description that will appear in the undo/redo/history palette.

Example 1:

This example starts an undo checkpoint.

```
Execute StartUndoCheckPoint Description('Adding some Text');
New Text 'Adding some Text';
...
Execute EndUndoCheckPoint;
```

Execute EndUndoCheckPoint

The Execute EndUndoCheckPoint command terminates a series of commands and groups them as one undoable action in the undo history, the undo history will be cleared when the group ends. Requires *Frame 7.2 or greater*. See the Users Guide for more information on the multiple undo/redo features.

Format:

```
Execute EndUndoCheckPoint [DocObject(docobjectvar)];
```

Table 39: Execute EndUndoCheckPoint Options

Option Name	Option Description
DocObject	The document object variable specifying the document. If not specified, the current active document is used.

Example 1:

This example starts an undo checkpoint, adds two lines of text and groups the new text as one undoable action in the undo history.

```
Execute StartUndoCheckPoint Description('Adding two lines of Text');
New Text 'Adding Text';
New Text 'Adding some more Text';
Execute EndUndoCheckPoint;
```

Execute ClearUndoHistory

The Execute ClearUndoHistory command clears the undo and redo history of the specified document. This also cancels any start group as well. Requires *Frame 7.2 or greater*. See the Users Guide for more information on the multiple undo/redo features.

Format:

```
Execute ClearUndoHistory [DocObject(docobjectvar)];
```

Table 40: Execute ClearUndoHistory Options

Option Name	Option Description
DocObject	The document object variable specifying the document. If not specified, the current active document is used.

Exec Script Command

The Exec Script command allows to start and run another script. This new script runs in its own data space, so there is no sharing of variables or data between script running the command and the script that is started. This is for standard scripts only

Format:

```
Exec Script FileName(fileNameString);
or
Exec Script ScriptText(stringValue);
```

Table 41: Exec Script Options

Option Name	Option Description
FileName	A string value containing the name of the script file.
ScriptText	A string value containing the text of the script. You must use one of FileName or ScriptText

Exec Wait Command

The Exec Wait command allows you to stop running for a specified amount of time. This allows you to stop execution of the current script so that some other program can run and, perhaps, produce some data that you need.

Format:

```
Exec Wait Seconds(NumberOfSecondsToWait);
or
Exec Wait MicroSeconds(NumberOfMicroSecondsToWait);
```

Table 42: Exec Wait Options

Option Name	Option Description
Seconds	An integer value specifying the number of seconds to wait.
MicroSeconds	An integer value specifying the number of micro-seconds to wait

Export Command

Export Object

The **Export Object** command exports a Frame Graphic Object file to a file on your hard disk.

Format:

```
Export Object (objVar) File(filename)
  [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
  [ReturnNativeError(nativeError)]
  [ExportOptions] See Table below.
```

Table 43: Export Object Options

Option Name	Option Description
Object (<i>Required</i>)	The Graphic object variable in the frame document that you wish to export.
File (<i>Required</i>)	The Filename for the exported object.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the export is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned. See “Return Status for Export Object” on page 36.
ReturnFilename	This option specifies a variable name to store a string containing the actual file name. This might be different than the specified file name if the ShowBrowser option is chosen.
ReturnNativeError	This option specifies a variable name to store an integer value that is usually the same as ErrorCode. This value is usually Zero for success, -42 for a system error, -43 for an option error or -44 for a cancelation. This value is usually Zero for success, -42 for a system error, -43 for an option error or -44 for a cancelation.

The following table lists the Export Object options. The default value for the option, if any, is in bold style text. For options that have True/False values, you may just use the option name to specify the True value (see import example).

IMPORTANT: Most of the time the default value for the export options will serve. You only need to specify an option and value if you want something to change.

IMPORTANT: The Export Object command is an unsupported FrameMaker FDK function.

Table 44: Export Options

Import Option	Value of Import Option
AlertUserAboutFailure	Alert the user if an unexpected condition happens. True or False
DontNotifyApiClients	Don't notify other Api clients about this export. By default, FrameMaker will notify Api clients when files are opened or closed (among many other events), if that client asks to be notified. In ElmScript, you do this by including an event procedure (NotePreOpenDoc, etc). This option lets you cancel that behavior by setting the value to True. True or False
FilterFormatId	The format id of the filter to use for this importation. FrameMaker will attempt to select a filter based on the file itself. In some case, it may select incorrectly. You may specify which filter to use by including this option on the import files command. The value is from the <code>formatId</code> in the maker.ini file. This option is available for FrameMaker 5.5 or greater. Default: a GIF will be assumed.
GraphicDpi	The Dpi (dots per inch) for the exported graphic.
ShowBrowser	This option tells FrameMaker to display the export dialog box to the user. True or False

Table 45: Return Status for Export Object

Status Text	Description
"ApiClientCanceledExport"	Export was canceled from an API client.
"BadExportDocId"	Bad Document Object for exported Object.
"BadExportFileName"	Invalid file name for this platform
"BadExportObjectId"	Bad Exported Object.
"BadExportScriptValue"	Bad export option.
"CanceledExportObjectNeedsDpi"	Missing Dpi
"ExportFileHasNewName"	Exported file has a new name.
"ExportFileNotWritable"	File was not writable
"ExportProductIsViewer"	Wrong Frame product. Export not allowed.
"FilterFailed"	Failure in the export filter.
"InsufficientMem"	Not enough memory to export this file.
"InvalidExportFilter"	Invalid export filter.
"InvalidTextRange"	Invalid Text Range.
"MissingExportScript"	Missing required option.
"UserCanceledExport"	Canceled by user

Example:

This example exports the first selected graphic in the document to a file called `c:\Images\gph1.gif`.

```

Set gvGraphicVar = FirstSelectedGraphicInDoc;
Export Object(gvGraphicVar) File('c:\Images\gph1.gif')
  FilterFormatId('GIF ') ReturnStatus(gvErrorList);
If ErrorCode not = 0
  Display 'Export Failure-'+ErrorCode+ ' Msg-'+ErrorMsg;
  Display 'Export Messages-'+gvErrorList;
EndIf

```

See also

“Import File” on page 56.

Find Commands

Find Member

Format:

```
Find Member(MemberValue)
      InList(Listvariable)
      [Binary]
      [Indirect(indList)]
      ReturnStatus(TrueFalsevarname)
      ReturnPos(positionvarname);
```

Table 46: Find Member (InList) Options

Option Name	Option Description
Member (<i>Required</i>)	The value to search for. This should match the type of data in the list.
InList	The name of the list variable in which to search.
Binary	This tells ElmScript to perform a binary search. For long lists (over 300 items), this will be faster than a sequential search, but it requires that the list be in ascending order. See “Sort command” on page 141. The default is to perform a sequential search.
indirect	This tells ElmScript to use an InList to indirectly indicate the order of the items to search. See “Sort command” on page 141.
ReturnStatus	Variable name. This is true if the string was found, false otherwise
ReturnPos	The position of the member in the list, if found.

Example 1:

This example searches the string list for the value 'STU'. It returns the position of the item in the list, which, in this case, is 2.

```
New StringList NewVar(gvList) value('ABC') Value('STU') value('XYZ');
Find Member ('STU') InList(gvList)
      ReturnStatus(gvFound) ReturnPos(gvPositionVar);
MsgBox 'Item found at position '+gvPositionVar+' in the list';
```

Example 2:

This example performs the same search as above, except it performs a binary search. The result is the same, but if the list had been very long (over 300 items) it would run faster.

```
New StringList NewVar(gvList) value('ABC') Value('STU') value('XYZ');
Find Member ('STU') InList(gvList) Binary
      ReturnStatus(gvFound) ReturnPos(gvPositionVar);
MsgBox 'Item found at position '+gvPositionVar+' in the list';
```

Example 3:

This example shows an indirect binary search. This gets names of all the fonts, sorts the list indirectly (it creates an integer list which points to the original list).

```
Set gvList = FontFamilyNames;
Sort List(gvList) case Indirect NewVar(gvSortedList);
Find Member('Helvetica') InList(gvList) Binary Indirect(gvSortedList);
```

Find String (In String)

The **Find String** command searches for a string in another string.

Format:

```
Find String(SearchString)
      InString (StringToSearch)
      [Start(integerexpression)]
      [NoCase WholeWord Backward Prefix Suffix]
      [ReturnPos(posvarname)] [ReturnLength(lengthVarName)]
      [ReturnStatus(TrueFalsevarname)]
      [ReturnString(strvarname)];
```

Table 47: Find String Options

Option Name	Option Description
String (Required)	The string to search for. This could be a string variable or a string constant or an expression combining the two.
InString (Required)	The string in which to search. This could be a string variable or a string constant or an expression combining the two.
Start	Character position in the string to start searching. The first character is 1. Default: start at beginning of string.
NoCase	Case insensitive (default-case sensitive)
WholeWord	Find whole word only
Backward	Search the string backward.
Prefix	Find the string only if it is the prefix of the InString string
Suffix	Find the string only if it is the suffix of the InString string
ReturnPos	Variable name. This will contain the starting position of the found string. (zero if not found)
ReturnLength	Variable name. This will contain the length of the found string. (zero if not found)
ReturnStatus	Variable name. This is true if the string was found, false otherwise
ReturnString	This is the value of the string found. This might be different from the search string, if NoCase was specified.

Example:

This example attempts to find the string 'The' in the string 'Now is the Time'.

```
Find String ('The') InString('Now is the Time') NoCase
      ReturnPos(gvPositionVar) ReturnStatus(gvFound) ReturnString(gvRetString);
```

After this command runs it, the 'gvPositionVar' variable contains the value 8, the 'gvFound' variable contains the value True, and the gvRetString variable contains the value 'the'.

Find String (In Frame Object)

Format:

```
Find String(SearchString)
    InObject (textobjectvariable) or InRange(rangevar)
    [Start(textlocation)]
    [NoCase WholeWord Prefix Suffix]
    [ScrollTo] [Center]
    ReturnRange(rangevarname)
    ReturnStatus(TrueFalsevarname)
    ReturnString(strvarname);
```

Table 48: Find String (InObject) Options

Option Name	Option Description
String (Required)	The string to search for. This could be a string variable or a string constant or an expression combining the two.
InRange	The text range in which to search.
InObject	The text object in which to search. This could be a paragraph, a textline or a flow. If neither the InRange nor InObject options are specified, the currently active document is used.
Start	Character position in the string to start searching. This should be valid text location.
NoCase	Case insensitive (default-case sensitive)
WholeWord	Find whole word only
Prefix	Find the string only if it is the prefix of the InString string
Suffix	Find the string only if it is the suffix of the InString string
ReturnRange	Variable name. This will contain the text range of the found string.
ReturnStatus	Variable name. This is true if the string was found, false otherwise
ReturnString	This is the value of the string found. This might be different from the search string, if NoCase was specified.
ScrollTo	Display the location of the find in the window.
Center	This is used with the ScrollTo option. It centers the found text in the document window.

Example:

This example searches the main flow of the currently active document for the word 'Text'.

```
Find String ('Text') InObject(MainFlowInDoc) NoCase Wholeword
    ReturnRange(rangevar) ReturnStat(found) ReturnString(strvar);
```

After this command runs, the 'rangevar' variable contains the text range, the 'found' variable contains the value True if it was found and False otherwise, and the strvar variable contains the value string value that was found.

Find TextRange

Format:

```
Find TextRange(trangevar) ScrollTo
```

Table 49: Find TextRange Options

Option Name	Option Description
TextRange	The text range to which to jump to.
ScrollTo	Display the location of the find in the window.

Example:

```
New TextRange NewVar(trangevar) Object(FirstPgfinDoc);
Find TextRange(trnagevar) ScrollTo;
```

These commands cause the document window to jump to the first paragraph in the document. Note that the first paragraph in the document is not necessarily the first paragraph in the flow.

Frame Find from text location

Format: This format is similar to the Frame UI Find command.

```
Find FromTextLoc[(TextLocVar)] searchobject
    [Case WholeWord Backward Wildcard NoWrap]
    [UseRegEx] FM 12 or greater
    [RegExFlavor(regexType)] FM 12 or greater
ReturnRange(textRangeVar)
ReturnStatus(TrueFalsevarname)
ReturnElementRange(eltRangeVar);
```

Table 50: Find FromTextLoc Options

Option Name	Option Description
FromTextLoc	A TextLoc data item that identifies the starting place for the search. If a textloc is not specified, then the search starts at the current cursor location. You MUST use the FromTextLoc option to indicate this method of the find whether you wish to specify a text location or not.
searchobject (Required)	The type of object for which to search. See “Frame Search Options” on page 41
Case	Case sensitive (default-case insensitive).
WholeWord	Find whole word only
Backward	Search for the string backward.
UseRegEx	Treat the text as a regular expression. <i>FM 12 or greater</i>
NoWrap	Do not wrap when the find reaches the location where the search began. <i>Frame 6.0 or higher</i>
Wildcard	Allow wildcards in search.

Table 50: Find FromTextLoc Options

Option Name	Option Description
RegexFlavor	The specifies the style of regular expression to use. This is only applicable if the UseRegex option is also used. The possible values are: UsePerl UseGrep UseEGrep <i>FM 12 or greater</i>
ReturnRange	Variable name. This will contain the text range of the found object.
ReturnStatus	Variable name. This is <code>true</code> if the string was found, <code>false</code> otherwise
ReturnElementRange	Variable name. This will contain the element range of the found object.

Table 51: Frame Search Options

Search type	Search target	Search value (if any)
String	Text	A text string identifying the string for which to search.
PgfTag	Paragraph Object	A string identifying a paragraph tag name.
ObjectStyleTag	Style Object	A string identifying an object style tag name. <i>FM 12 or greater</i>
CharTag		
Table	Table Anchor	A text string identifying a table tag. If no value is given, then it finds next table of any type.
Marker	Marker Anchor	A text string containing the name of the marker type to find. If no value is given, then it finds the next marker of any type.
MarkerText	Marker Anchor	A text string containing the marker text to find.
XRef	XRef	A text string containing the name of the XRef format for which to search. If no value is given, then it finds the next XRef of any type
Variable	Variable	A text string containing the format name of the variable for which to search. If no value is given, then it finds the next Variable of any type
Element	Element	A stringlist optionally containing three items for which to search. The first is the element name. The second is an attribute name and the third is an attribute value. Any of these values may be empty.
InCond	Text	StringList containing the names of one or more condition tag names.
NotInCond	Text	StringList containing the names of one or more condition tag names.
UnresolvedXRef	XRef	No Value. This finds the next unresolved cross-reference.
TextInset	TextInset	No Value. This finds the next text inset of any type.
UnresolvedTextInset	TextInset	No Value. This finds the next unresolved text inset.
Pub	Text	No Value. This finds the next publisher (Macintosh only).
AFrame	AnchoredFrame	No Value. This finds the next anchored frame.
Footnote	Footnote	No Value. This finds the next footnote.
AutoHyphen	Text	No Value. Finds the next autohyphen text

Table 51: Frame Search Options

Search type	Search target	Search value (if any)
Rubi	Text	No Value. Finds the next rubi.
PgfFormatOverride	Text	No Value. Finds the next paragraph with an override. <i>FM 10 or greater</i>
CharFormatOverride	Text	No Value. Finds the next next char with an override. <i>FM 10 or greater</i>
TableFormatOverride	Text	No Value. Finds the next next table with an override. <i>FM 10 or greater</i>
FontFamily	Text	Font family name
CombinedFontFamily	Text	Combined font name
You may choose one or more of the following properties in the same search.		
CharTag	Text	Name of the character tag
FontSize	Text	Metric font size
FontAngle	Text	Font angle name
FontWeight	Text	Font weight name
FontVariation	Text	Font variation name
Color	Text	Color name
Spread	Text	Metric spread value
Stretch	Text	Metric stretch value
Language	Text	Language value, see “Language” on page 166
Underlining	Text	Underlining value, see “Underlining” on page 166
Overline	Text	Overline value, see “Overline” on page 166
StrikeThrough	Text	StrikeThrough value, see “Strikethrough” on page 166
ChangeBar	Text	Change bar value, see “ChangeBar” on page 165
Capitalization	Text	Capitalization value, see “Capitalization” on page 165
Position	Text	Position value, see “Position” on page 166

Example 1:

This example attempts to find the next 'Procedure' element that has a 'type' attribute with the value 'Specific'.

```

. . .
NEW SelectList NewVar(elist) Value('Procedure')
    Value('Type')
    Value('Specific');
Find FromTextLoc Element(elist) ReturnRange(trange)
    ReturnStatus(stat) ReturnElementRange(erange);
. . .

```

Example 2:

This example attempts to find the next variable that has a variable format of 'MyVar'.

```

Find FromTextLoc Variable('MyVar') ReturnRange(trange)
    ReturnStatus(stat);

```

Example 3:

This example attempts to find the next text location that has text fontsize of 12 points and bold weight.

```
Find FromTextLoc FontSize(12) FontWeight('Bold') ReturnRange(trange)
ReturnStatus(stat);
```

Function Command

The following shows the format of the Function declaration command.

Format:

```
Function functionName [using [ByRef ]arg1[ [ByRef ]arg2]...[ [ByRef ]argN]];
. . .
EndFunction
```

In between the Function and EndFunction lines there can be any number of commands.

Table 52: Function Options

Option Name	Option Description
functionName (Required)	The name of the subroutine. This name has to be unique within the script. You cannot have two subroutines with the same name in the same script file.
using	An optional filler word to enhance readability.
argI	The name of an argument. These are names (identifiers) of your choosing. Each name represents the name of a variable in the parameter data space for this function. These are used as variable names inside the subroutine. The order of these names determine the order of the arguments required when the function is called. The Args array is an alternate way to access the arguments.
EndFunction (Required)	The command that terminates a function.

Generate BookFiles

Generate Bookfiles command generates files for a book. It performs the same operation as choosing Generate from the book File menu. The book and its generated files must be set up before you call this function

Format

```
Generate BookFiles [BookObject (bookvar)] [{Interactive Visible}]
```

Table 53: Generate BookFiles Options

Option Name	Option Description
BookObject	The book object variable specifying the book. If not specified, the current active book is used.
Interactive	Specifies whether to display warnings and messages to the user (Default=False).
Visible	Specifies whether to display the generated files (Default=True).

Example

The following code generates files for the currently active book:

```

. . .
Generate BookFiles Visible;
. . .

```

Get Commands

Get BookConditionalBuildExpression(*FM 2015 or greater*)

The **Get BookConditionalBuildExpression** command gets the build expression for the name given and for the specified book. These settings are returns in a DataRecord data type

Format

```

Get BookConditionalBuildExpression [BookObject(bookObj)] ExprName(condName)
NewVar(Expr);

```

Table 54: Get BookConditionalBuildExpression Options

Option Name	Option Description
BookObject	The book object. The default value is the current book..
ExprName	Specifies the name of the build expression to retrieve.
NewVar	Specifies the name of the variable to put the build expression

Example 1:

The following code gets the conditional settings for the currently active book: and checks the show status.

```

. . .
Get BookConditionalBuildExpression ExprName('MyBuildExpression') NewVar(gvExpr);
Display 'The build expression for MyBuildExpression is'+gvExpr;
. . .

```

Get BookConditionalSettings(*FM 2015 or greater*)

The **Get BookConditionalSettings** command gets the conditional settings for the specified book. These settings are returns in a DataRecord data type

Format

```

Get BookConditionalSettings [BookObject(bookObj)] NewVar(condSettings);

```

Table 55: Get BookConditionalSettings Options

Option Name	Option Description
BookObject	The book object. The default value is the current book..
NewVar	<p>Specifies the name of the variable to put the results (DataRecord). The values in the data record are as follows:</p> <p>ShowState (ShowAll, ShowAsPerConditions, ShowAsPerExpression)</p> <p>ShowConditions-A StringList of condition names in the show state</p> <p>HideConditions-A StringList of condition names in the hide state</p> <p>ConditionalExpressions-A StringList of all condition names</p> <p>ActiveConditionalExpression-String value with the active condition expression tag</p> <p>ShowConditionIndicators-Show/Hide condition indicators</p> <p>ApplyConditionalSettingsToViewOnlyDoc-True, Apply the conditions to view only documents</p> <p>ApplyConditionalSettingsToNestedBooks-True, Apply the conditions to nested books</p> <p>ApplyConditionalSettingsShowBookErrorLog-True, Show errors in the book error log.</p>

Example 1:

The following code gets the conditional settings for the currently active book: and checks the show status.

```

. . .
Get BookConditionalSettings NewVar(gvCondSettings);
if gvCondSettings.ShowState=ShowAll
    Display 'Settings are all on';
EndIf
. . .

```

Get Member

The **Get Member** command gets an individual member from a list data item.

Format

```
Get Member [Number(membernumber)] From(listvar) NewVar(varname);
```

Note: This command is now obsolete. It is kept for backward compatibility with previous versions. You can access members easier and more efficiently using the index operator ([]).

Table 56: Get Member Options

Option Name	Option Description
Number	The integer number of the member in the list. The first member is 1.
From (Required)	Specifies the list variable name.
NewVar	Specifies the name of the variable to put the member.

Example 1:

The following code gets the specified member from the string list:

```

. . .
New StringList NewVar(gvStringList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Get Member Number(2) From(gvStringList) NewVar(gvString);
Display gvString;
. . .

```

The value of `gvString` will be 'Oranges'.

Example 2:

This is the same as above except this uses the indexing operator instead of the Get member command:

```

. . .
New StringList NewVar(gvStringList)
    Value('Apples') Value('Oranges') Value('Peaches');
....
Display gvStringList[2];
. . .

```

The value displayed will be 'Oranges'.

Get Object

The **Get Object** command gets a Frame 'named' object or an object that has a Unique ID. A certain type of objects in a FrameMaker system have unique names. You may use this command to get the object variable for one of these objects by providing the name. Other objects have unique Id properties (**Unique** property, see the Object Properties in the next chapter). You can use the second format of this command to retrieve those objects.

Format

```

Get Object [DocObject(docvar)] Type(objecttype) Name(objectname) NewVar(varname);
or
Get Object [DocObject(docvar)] Type(objecttype) Unique(uniqueId) NewVar(varname);

```

Table 57: Get Object Options

Option Name	Option Description
DocObject	The document object variable. If not specified the currently active document is used.
Name	Specifies name of the Object. You must use this option or the Unique option to identify the object to get.
Unique	Specifies the unique Id of the object. You get this from the Unique property of any object that has this property. You must use this option or the Name option to identify the object to get.

Table 57: Get Object Options

Option Name	Option Description
Type (Required)	The object type to look for. If the name option is used, then choose the Type from the following table (“Object Types (for Get Object by Name Command)” on page 47). If the Unique option is chosen, then choose the object type from the next table (“Object Types (for Get Object by UniqueId Command)” on page 47). You can specify this option by typing in the code exactly as in the Object Type column of the following tables or you can specify it as a string value.
NewVar (Required)	Specifies the name of the variable to put the object.

Table 58: Object Types (for Get Object by Name Command)

Object Type	Object Description
CharFmt	Character Format.
Color	Color.
Command	Menu item command.
CondFmt	Condition Format
ElementDef	Element Definition object
GraphicsFmt	Graphics Formatobject
MarkerType	Marker type object
MasterPage	Master page
Menu	Menu
MenuItemSeparator	Menu Item Separator
Pgf	Paragraph Format
PgfFmt	Paragraph Format
RefPage	Reference page
RulingFmt	Ruling Format
TblFmt	Table Format
UnanchoredFrame	Unanchored Frame (reference frame)
VarFmt	Variable Format
XRefFmt	Cross reference format

Table 59: Object Types (for Get Object by UniqueId Command)

Object Type	Object Description
AFrame	Anchored Frame Object.
Arc	Arc Graphic Object.
BookComponent	Book Component. Object

Table 59: Object Types (for Get Object by UniqueId Command)

Object Type	Object Description
Cell	Table Cell
Element	Element Object.
Ellipse	Ellipse Graphic Object.
Fn	Footnote Object.
Group	Group Graphic Object.
Inset	Cross reference format
Line	Line Graphic Object.
Marker	Marker Object
Math	Math Graphic Object.
MathML	MathML Graphic Object. <i>FM 12 or greater.</i>
Pgf	Paragraph
Polygon	Polygon Graphic Object.
PolyLine	PolyLine Graphic Object
Rectangle	Rectangle Graphic Object.
RoundRect	Round Rectangle Graphic Object.
SubCol	Sub Column Object.
Tbl	Table
TextFrame	Text Frame Object.
TextLine	Text Line Graphic Object.
TiApiClient	Text Inset ApiClient
TiFlow	Text Inset Flow
TiText	Text Inset Text
TiTextTable	Text Inset Table
UnanchoredFrame	Unanchored Frame
Var	FrameMaker Variable Object
XRef	Cross-Reference Object.

Example

The following code gets the object variable of the 'Heading2' paragraph format:

```

. . .
Get Object Name('Heading2') Type (PgFFmt) NewVar(strmem);
. . .

```

Example

The following code gets the object variable of the 'Heading2' paragraph format:

```

. . .
Set gvBkComp = ActiveBook.FirstBookComponent
Get Object Name('Heading2') Type (PgFmt) NewVar(strmem);
. . .

```

See also

“New Named Objects” on page 87.

Get String

The **Get String** command retrieves a sub string from another string. The string can be modified as it is retrieved (e.g. uppercase). You can use this command to perform many string operations.

Format:

```

Get String FromString(TargetString)
    [StartPos (startcharacterposition)]
    [EndPos (lastcharacterposition)]
    [RemoveLeading('LeadingCharactersToRemove')]
    [RemoveTrailing('TrailingCharactersToRemove')]
    [RemoveChars('CharactersToRemove')]
    [UpperCase] [LowerCase]
    [Reverse]
    [ReplaceFirst(Str1) With(Str2)]
    [ReplaceLast(Str1) With(Str2)]
    [ReplaceAll(Str1) With(Str2)]
    [PlatformToFrame]
    [FrameToPlatform]
    NewVar(varname)

```

Table 60: Get String Options

Option Name	Option Description
FromString (<i>Required</i>)	The string to search. This could be a string variable or a string constant or an expression combining the two.
StartPos	The starting character position.
EndPos	The ending character position
RemoveLeading	A string containing a list of characters to remove from the beginning of the retrieved string.
RemoveTrailing	A string containing a list of characters to remove from the end of the retrieved string.
RemoveChars	A string containing a list of characters to remove from the retrieved string.
UpperCase	Converts the characters to upper case.
LowerCase	Converts the characters to lower case.
ReplaceFirst	The first occurrence of this string is located and replaced with the corresponding <code>With</code> string. You may have zero or more of these sets of options.
ReplaceLast	The last occurrence of this string is located and replaced with the corresponding <code>With</code> string. You may have zero or more of these sets of options.
ReplaceAll	All occurrences of this string are located and replaced with the corresponding <code>With</code> string. You may have zero or more of these sets of options.
With	This identifies a replacement string for a preceding <code>ReplaceFirst</code> , <code>ReplaceLast</code> or <code>ReplaceAll</code> option.
Reverse	Reverses all the characters in the string
FrameToPlatform	Converts the characters from the <code>FrameMaker</code> character set to the platform character set. If the original string is not in the <code>FrameMaker</code> character set, the results are unpredictable.
PlatformToFrame	Converts the characters from the <code>Platform</code> character set to the <code>FrameMaker</code> character set. If the original string is not in the <code>Platform</code> character set, the results are unpredictable.
NewVar (<i>Required</i>)	This is the returned string.

The order of the string operations is the same order as they are listed above, if they are specified. The `StartPos` and `EndPos` create a substring first (if specified), then the `RemoveLeading`, `RemoveTrailing` and `RemoveChars` occur next, followed by the `Upper` and/or `Lower` casing. The `ReplaceFirst`, `ReplaceLast`, `ReplaceAll` groups go next, followed by the `Reverse` option. Finally, the `FrameToPlatform` or `PlatformToFrame` options are applied. If any of these options are not specified, then that step is skipped.

Example 1

The following code pulls a substring (characters 6 through 9) from the original string and changes the characters to uppercase.

```

. . .
Set gvTestString = 'This is a test string';
Get String FromString(gvTestString) NewVar(gvExtractedString)
  StartPos(6) EndPos(9) UpperCase;

```

The value in the `gvExtractedString` variable would be 'IS A'.

Example 2

The following code pulls a substring (starting with character 3) from the original string, and removes the spaces from the string.

```

. . .
Set gvTestString = 'This is a test string';
Get String FromString(gvTestString) NewVar(gvExtractedString)
  StartPos(3) RemoveChars(' ');

```

The value in the gvExtractedString variable would be 'isisateststring'.

Example 3

The following code pulls a substring (starting with character 3) from the original string, and removes the spaces from the string.

```

. . .
Set gvTestString = '   my name   ';
Get String FromString(gvTestString) NewVar(gvExtractedString)
  RemoveLeading(' ') RemoveTrailing(' ') UpperCase;

```

The value in the gvExtractedString variable would be 'MY NAME'.

Example 4

The following code replaces the first AA string with BB, replaces all periods(.) with commas(,) and the QAZZAQ with ??.

```

. . .
Set gvTestString = 'AAAA1234.45QAZZAQ';
Get String FromString(gvTestString) NewVar(gvExtractedString)
  ReplaceFirst('AA') With('BB')
  ReplaceAll('.') With(',')
  ReplaceAll('QAZZAQ' With('??'));

```

The value in the gvExtractedString variable would be 'BBAA1234,45??'.

Get TextList

The **Get TextList** command gets a list of text items from a FrameMaker text object. A text object is a paragraph or textline. FrameMaker stores text and its properties in a text item list (see data type description).

Format

```

Get TextList {InObject(objvar) or InRange(rangevar)}
  NewVar(varname) [listoftextitemtypes] or ItemTypeList(strList);

```

Table 61: Get TextList Options

Option Name	Option Description
InObject	The object variable (paragraph or textline). This option or the InRange option is required.
InRange	A text range.
listoftextitemtypes	Specifies the list text types to retrieve (see the following table).
ItemTypeList	Specifies the list text types to retrieve as a StringList variable with each string in the list the name of the Text Item type. This option allows you to dynamically specify the text items instead of hard-coding them into the command.
NewVar (Required)	Specifies the name of the variable to put the result.

Table 62: Text Item Type List

Type Item Type	Text Item Type Description	Data Type (TextData)
CharPropsChange	A character property has change.	StringList of changes, see “Character Property Changes” on page 53.
ElementBegin	Start of a container structural element	Element Object
ElementEnd	End of a container structural element	Element Object
ElemPrefixBegin	Start of an element’s prefix	Element Object
ElemPrefixEnd	End of an element’s prefix	Element Object
ElemSuffixBegin	Start of an element’s suffix	Element Object
ElemSuffixEnd	End of an element’s suffix	Element Object
FlowBegin	Start of Flow Item	Flow Object
FlowEnd	End of Flow Item	Flow Object
FnAnchor	Footnote Anchor Item	Footnote Object
FrameAnchor	Anchored Frame Item	Anchored Frame Object
LineBegin	Start of new Line Item	0
LineEnd	End of Line Item	End of line type 0-normal 1-Hard end of line 2-hyphenation
MarkerAnchor	Marker Anchor Item	Marker Object
PageBegin	Start of page Item	Page Object
PageEnd	End of page Item	Page Object
PgfBegin	Start of Paragraph Item.	Paragraph Object
PgfEnd	End of Paragraph Item.	Paragraph Object
RubiCompositeBegin	Start of a Rubi Composite (and implicitly the start of oyamoji Text	Rubi Object
RubiCompositeEnd	End of a Rubi Composite.	Rubi Object
RubiTextBegin	Start of Rubi Text.	Rubi Object
RubiTextEnd	End of Rubi Text.	Rubi Object
String	A string fragment Item.	String value
SubColBegin	Start of sub column Item	Sub column Object
SubColEnd	End of sub column Item	Sub column Object
TblAnchor	Table Anchor Item	Table Object
TextFrameBegin	Start of Text Frame Item.	Textframe Object
TextFrameEnd	End of Text Frame Item.	Textframe Object
TextInsetBegin	Start of Text Inset Item	Text Inset Object
TextInsetEnd	End of Text Inset Item	Text Inset Object

Table 62: Text Item Type List

Type Item Type	Text Item Type Description	Data Type (TextData)
TextObjId	Object that the offsets of all text items are relative to.	Pgf Object or TextLine Object
VarBegin	Start of variable Item	Variable Object
VarEnd	End of variable Item	Variable Object
XRefBegin	Start of Cross-reference Item	Cross-Reference Object (XRef)
XRefEnd	End of Cross-reference Item	Cross-Reference Object (XRef)

Table 63: Character Property Changes

CharPropsChange	Description
ANGLE	The font angle has changed
CAPITALIZATION	The capitalization has changed
CHANGEBAR	The change bars have changed
CHARTAG	The Character format has changed
COLOR	The color has changed
CONDITIONTAG	The condition tag has changed
ENCODING	The text encoding has changed
FAMILY	The font family has changed
IFF	An internal flag having to do with asian text input. If there is a non-zero value for this flag, a front end process is controlling that text; you should not modify the associated ite,.
KERNX	The kern-x characteristic has changed
KERNY	The kern-y characteristic has changed
LANGUAGE	The character language has changed
OUTLINE	The outline characteristic has changed
OVERLINE	The overline characteristic has changed
PAIRKERN	The pair kerning has changed
POSITION	The character position has changed
SHADOW	The shadow characteristic has changed
SIZE	The font size has changed
SPREAD	The font spread has changed
STRETCH	The font stretch value has changed
STRIKETHROUGH	The strikethrough characteristic has changed
TSUME	The Tsume setting has changed
UNDERLINING	The underlining has changed

Table 63: Character Property Changes

CharPropsChange	Description
VARIATION	The font variation has changed
WEIGHT	The font weight has changed

Example

The following code gets a list text items from the first paragraph in the currently active document (getting the text and the paragraph begin and end tags), then it displays then on the FrameMaker console.

```

. . .
Get TextList InObject(FirstPgInDoc) NewVar(tlist)
    String PgfBegin PgfEnd;
Loop LoopVar(idx) InitVal(1) Incr(1) While(idx<=tlist.count)
    Get Member Number(idx) From(tlist) NewVar(titem);
    Write console 'Text Item#'+idx+' Offset='+titem.TextOffset+
        ' Data='+titem.TextData;
EndLoop
. . .

```

Example

The following code is the same as the previous example except that it uses a TextTypeList to specify the kinds of text item types to retrieve. It gets a list text items from the first paragraph in the currently active document (getting the text and the paragraph begin and end tags), then it displays then on the FrameMaker console.

```

. . .
New StringList NewVar(gvTextItems) Value('PgfBegin') Value('PgfEnd') Value('String');
Get TextList InObject(FirstPgInDoc) NewVar(tlist) ItemTypeList(gvTextItems);
Loop LoopVar(idx) InitVal(1) Incr(1) While(idx<=tlist.count)
    Get Member Number(idx) From(tlist) NewVar(titem);
    Write console 'Text Item#'+idx+' Offset='+titem.TextOffset+
        ' Data='+titem.TextData;
EndLoop
. . .

```

See also

“New Text” on page 99.

Get TextProperties

The **Get TextProperties** command gets a list of the text properties at the specified text location.

Format

```
Get TextProperties [DocObject(docvar) or TextLoc(textlocvar)] NewVar(varname);
```

Table 64: Get TextProperties Options

Option Name	Option Description
DocObject	The Document object variable.

Table 64: Get TextProperties Options

Option Name	Option Description
TextLoc	The Text location. If the text location is not specified, ElmScript uses the current insertion point of the DocObject option. If neither is specified, the insertion point of the currently active document is used.
NewVar (<i>Required</i>)	Specifies the name of the variable.

Example

The following code gets a list text properties from the insertion point in the currently active document.

```

. . .
Get TextProperties NewVar(tprops)
. . .

```

See also

“Apply TextProperties” on page 8.

GlobalVar command

The **GlobalVar** command creates or updates a variable in the global data space for a script. You may create or update more than one variable with one command.

Format:

```

GlobalVar VarName1[(expression)][ VarName2[(expression2)] . . .
{ VarNameN[(expressionN)]};

```

Table 65: GlobalVar Options

Option Name	Option Description
VarNameI (<i>Required</i>)	The name of an existing variable or a new variable name. ElmScript will create a new variable in the global data space if it doesn't already exist. This will create a new global variable (if necessary) regardless of the value in the DeclareVarMode session variable.
expressionI	The expression to compute. This can be any valid arithmetic or string expression, using variables and/or properties. If the expression is not specified, then the variable is assigned the value of NULL.

If, Else, ElseIf, EndIf

The If command allows you to selectively execute a list of commands based on the evaluation of an expression.

Format:

```

If expression
  command list;
[ElseIf expression
  command list;]
[Else
  command list;]
EndIf

```

If the expression evaluates to a `True` value, then the first set of commands are executed. If the expression evaluates to `False`, then the second set of commands are executed (if present). The expression can be any type of valid expression. A numerical value is `False` if the value is zero, otherwise it is `True`. A String expression is `False` if the string is empty, otherwise it is `True`. If an object is zero, the expression is `False`, otherwise it is `true`. For list variables, if there are items in the list then it is `True`; if it is an empty list (no members) then it is `False`.

Be sure to use the `EndIf` at the end of the command list otherwise the command list under the `If` command goes on until the end of the subroutine or script.

Example:

```
DialogBox Type(Int) Title('Enter a non-zero number') NewVar(gvTestNumber)
If gvTestNumber = 0
    MsgBox 'Invalid number entered!!!';
Endif
```

Import Commands

Import File

The `Import File` command imports a file into an open document.

Format:

```
Import File File(filename) [DocObject(docvar)]
    [TextLoc(textlocation)] [NewVar(textinsetobjectvar)]
    [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
    [ReturnNativeError(nativeError)]
    [ImportOptions] See Table below.
```

Table 66: Import File Command Options

Option Name	Option Description
DocObject	The Document object variable of the target document. If not specified the imported document will go into the currently active document.
File (Required)	Filename of document to import.
TextLoc	The Text location. If the text location is not specified, ElmScript uses the current insertion point of the DocObject option. If neither is specified, the insertion point of the currently active document is used.
NewVar	Specifies the name of the variable to put the object of the text inset if present. This occurs only if you import by reference a text inset object.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the import is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned. See “Return Status Values for Import” on page 60.
ReturnFilename	This option specifies a variable name to store a string containing the actual file name. This might be different than the specified file name if the ShowBrowser option is chosen.
ReturnNativeError	This option specifies a variable name to store an integer value that is usually the same as ErrorCode. This value is usually Zero for success, -42 for a system error, -43 for an option error or -44 for a cancellation.

The following table lists the Import file options. The default value for the option, if any, is in bold style text. For options that have True/False values, you may just use the option name to specify the True value (see import example).

IMPORTANT: Most of the time the default value for the import options will serve. You only need to specify an option and value if you want something to change.

Table 67: Import File Options

Import Option	Value of Import Option
AlertUserAboutFailure	Alert the user if an unexpected condition happens. True or False
CellSeparator	If the <code>FileIsText</code> option has the value <code>DoImportAsTable</code> , this option specifies the separator used to parse the text into table cells. The default value is <code>Tab</code> .
DisallowDoc	Disallow the importation of FrameMaker binary documents. True or False
DisallowFilterTypes	Disallow the importation of documents via filters. True or False
DisallowGraphicTypes	Disallow the importation of graphic files. True or False
DisallowMacEditions	Disallow the importation of Macintosh Editions documents. True or False
DisallowMif	Disallow the importation of FrameMaker mif documents. True or False
DisallowPlainText	Disallow the importation of plain text files. True or False
DisallowSgml	Disallow the importation of SGML files. True or False
DisallowXml	Disallow the importation of Xml files. FM 7.0 or greater True or False
DitaMaxRefLevels	Specifies the number of reference levels to be opened while opening a DITA file. FM 2017 or greater
DoNotLockFile	Specifies whether to lock the file or not. FM 2017 or greater True or False
DontNotifyApiClients	Don't notify other Api clients about this importation. By default, FrameMaker will notify Api clients when files are opened or closed (among many other events), if that client asks to be notified. In ElmScript, you do this by including an event procedure (<code>NotePreOpenDoc</code> , etc). This option lets you cancel that behavior by setting the value to <code>True</code> . True or False
FileIsGraphic	The imported file is a graphics file. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (ShowDialog). OK or Cancel or DoShowDialog

Table 67: Import File Options

Import Option	Value of Import Option
FileIsMakerDoc	The imported file is a FrameMaker binary or mif file. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
FileIsSgmlDoc	The imported file is an SGML file. You can import it, cancel the operation or let the user decide. OK or Cancel or DoShowDialog
FileIsXmlDoc	The imported file is an Xml file. You can import it, cancel the operation or let the user decide. FM 7.0 or greater OK or Cancel or DoShowDialog
FileIsText	The imported file is a text file. You may choose one of the following options to format the imported text. Choose the <code>TextFileEolIsEop</code> option (default) to convert each end of line into a paragraph break. Use the <code>TextFileEolIsNotEop</code> option to import without converting the text lines into paragraph breaks. Use the <code>DoImportAsTable</code> option to import the text into a table. Use the <code>Cancel</code> option to stop the importation altogether, TextFileEolIsEOP , <code>TextFileEolIsNotEop</code> , <code>DoImportAsTable</code> , <code>DoShowDialog</code> or <code>Cancel</code>
FilterFormatId	The format id of the filter to use for this importation. FrameMaker will attempt to select a filter based on the file itself. In some case, it may select incorrectly. You may specify which filter to use by including this option on the import files command. The value is from the <code>formatId</code> in the <code>maker.ini</code> file. This option is available for FrameMaker 5.5 or greater. Default: Have FrameMaker select the filter.
FitGraphicInSelectedRect	Attempt to fit the imported graphic in the selected graphic frame. True or False
ForceImportAsText	Import the document as text only. Not all files can be imported as text (e.g. FrameMaker binary files). True or False
FormatImportedText	The option specifies how the imported text will be formatted. You may use the formatting from the enclosing document (the default), format it as plain text (<code>PlainText</code>) or use the imported document (<code>SourceDoc</code>). EnclosingDoc or <code>PlainText</code> or <code>SourceDoc</code>
GraphicDpi	The Dpi (dots per inch) for the imported graphic.
HowToImport	The method used to import the file. This may be done by Reference (default), by Copy or by asking the user to choose. You may also specify the reference or copy options by using the name of the value (<code>DoByRef</code> or <code>DoByCopy</code>) as a shortcut. DoByRef or <code>DoByCopy</code> or <code>DoUserChoice</code>
ImportAsType	The format of the file to import. <code>TypeBinary</code> - A FrameMaker binary file (standard FrameMaker file) <code>TypeMif</code> - A MIF file <code>TypeText</code> - A Text file <code>TypeSgml</code> - An Sgml File <code>TypeXml</code> - An Xml File (FM 7.0 or greater) Default: Frame determines the type.
ImportFlowPageSpace	This option specifies where to search for the flow to import. This is only applicable if the <code>UseMainFlow</code> option is <code>False</code> . The actual flow tag to search is specified by the <code>ImportFlowTag</code> option. You may ask FrameMaker to search the Body pages (<code>BodyPage</code> , the default) or the reference pages (<code>ReferencePage</code>). BodyPage or <code>ReferencePage</code>

Table 67: Import File Options

Import Option	Value of Import Option
ImportFlowTag	This option specifies the flow tag to search for, when the UseMainFlow option is False. Value='string expression', default NONE
ImportTblTag	This option identifies the table tag to use when importing a text file into a table. This is only applicable when the FileIsText option is set to DoImportAsTable. Value is 'string expression' default NONE
InsetElementDef	FM 11 or greater
LeaveHeadingRowsEmpty	This option lets you leave the table's header cells empty when importing text files into tables. Set this option to True to keep the header rows empty. This is only applicable when the FileIsText option is set to DoImportAsTable. True or False
ManualUpdate	Update an inset manually. True or False
MappingFile	The mapping file FM 2015 or greater Value is 'string expression' default NONE
NumCellSeparators	This option lets you specify the number of spaces to require when the CellSeparator option is set to space. This is only applicable when the FileIsText option is set to DoImportAsTable. Value is an integer expression, default 1
NumColumns	This option lets you specify the number of columns when importing a file as text into a table when the TreatParaAsRow option is set to False. This is only applicable when the FileIsText option is set to DoImportAsTable. Value is an integer expression, default 1
PDFPageNum	The page number of the PDF file to import. <i>For FrameMaker 7.1 and greater.</i>
RasterDpi	Undocumented Metric option.
RasterImageHeight	Undocumented Metric option.
RasterImageWidth	Undocumented Metric option.
RemoveManualPageBreaks	Remove the manual page breaks from the imported document. The option is applicable only if FormatImportedText option is set to EnclosingDoc. True or False
RemoveOverrides	Remove the format overrides from the imported document. The option is applicable only if FormatImportedText option is set to EnclosingDoc. If set to True FrameMaker will remove the format overrides. True or False
ShowBrowser	This option tells FrameMaker to display the importation dialog box to the user. True or False
ShowRasterDpiDialog	Undocumented Integer option.
SgmlImportApplication	This option specifies the name of the SGML application to use when importing an SGML file. This takes precedence over any other SGML application specified.
StructuredImportApplication	This option specifies the name of the structure application to use when importing an XML or SGML file. This takes precedence over any other structure application specified.

Table 67: Import File Options

Import Option	Value of Import Option
TblNumHeadingRows	This option lets you specify the number of heading rows when importing text files into tables. This is only applicable when the <code>FileIsText</code> option is set to <code>DoImportAsTable</code> . (integer expression) <i>optional default 1</i>
TextInsetName	The name of the text inset default NONE
TreatParaAsRow	Set this option to <code>True</code> to have FrameMaker convert each line into a row of table cells and <code>False</code> to convert each line into a separate cell instead. This is only applicable when the <code>FileIsText</code> option is set to <code>DoImportAsTable</code> . True or False
UseMainFlow	Get the text from the Main Flow of the imported document. If set to <code>False</code> FrameMaker will not get the text from the imported documents main flow. True or False
UseHTTP	

Table 68: Return Status Values for Import

Status Text	Description
"BadEnclosingDocId"	There is no open document with the specified Object.
"BadImportFileName"	The specified source filename is invalid.
"BadImportFileType"	The Import options specified a file type different from the source file's actual type.
"BadImportScriptValue"	The Import options contained an invalid property value.
"BadTextFileTypeHint"	The file was a text file, and the string in <code>FilterFormatId</code> was not a valid value.
"CancelFileDoc"	The file is a FASL file, so the user or the options canceled the Import operation.
"CancelFileFilterable"	The source file is a filterable file, so the user or the options canceled the Import operation.
"CancelFileGraphic"	The source file is a graphic, so the user or the Import options canceled the Import operation.
"CancelFileIsMacEdition"	The source file is a Macintosh Edition, so the Import options canceled the Import operation.
"CancelFileMIF"	The source file is a MIF file, so the user or the options canceled the Import operation.
"CancelFileSgml"	The file is an SGML document, so the user or the options canceled the Import operation.
"CancelFileText"	The file is text, so the user or the Import options canceled the Import operation.
"CancelFileXml"	The file is an XML document, so the user or the options canceled the Import operation.
"CancelImportBrowser"	
"CantForceImportAsText"	
"DisallowedImportType"	The source file's type disallowed by script.
"FlowUnstructured"	
"ImportedByCopy"	The source file was imported by copy.
"ImportedFilteredFile"	The source file was filtered.

Table 68: Return Status Values for Import

Status Text	Description
"ImportedGraphicFile"	The source file is a graphics file.
"ImportedMacEdition"	
"ImportedMakerDoc"	The source file is a FASL file.
"ImportedMIF"	The source file is a MIF file.
"ImportedSgmlDoc"	The source file is an SGML document.
"ImportedText"	The source file is a text file.
"ImportedTextTable"	The source file is a text file, which was imported into a table.
"ImportedXmlDoc"	The source file is an XML document.
"ImportFileNotReadable"	The specified source file is unreadable.
"InsertionPointInFootnote"	The insertion point was in a footnote and the import options specified to import the file as a table, so the file could not be imported.
"InsertionPointInTableCell"	The insertion point was in a table cell and the import options specified to import the file as a table, so the file could not be imported.
"InsertionPointNotInText"	The insertion point in the enclosing document is not in text.
"InsufficientMemory"	There is insufficient memory to import the source file.
"MissingImportScript"	Internal Error.
"NoFlowWithSpecifiedName"	The script specified a flow name that does not exist.
"NoMainFlow"	The script specified to import the main flow, but the source file does not have a main flow.
"UserCanceledImport"	The user canceled the Import operation.

Example 1:

The following example imports a document, by copy method, at the insertion point of the currently active document. Don't allow any plain text or mif documents to be processed.

```

Import File File('testimport.fm')
  ByCopy
  DisallowPlainText
  DisallowMif;

```

Example 2:

The following example imports a text document, by copy method, into a Frame table at the insertion point of the currently active document. Don't allow any plain text or mif documents to be processed.

```

Set fname = 'C:\ElmScript\docs\test.txt';
New string NewVar(TABCH) intvalue(8);
Set errorcode = 0;
Import File File(fname)
    HowToImport(DoByCopy)
    FileIsText(DoImportAsTable)
    AlertUserAboutFailure(False)
    CellSeparator(TABCH)
    ImportTblTag('Format A')
    LeaveHeadingRowsEmpty(False)
    NumCellSeparators(1)
    TreatParaAsRow(True)
    TblNumHeadingRows(1)
    ReturnStatus(gvErrorList);
If ErrorCode not = 0
    Display 'Import Failure-'+ErrorCode+ ' Msg-'+ErrorMsg;
    Display 'Import Messages-'+gvErrorList;
EndIf
    
```

See also

“Import Formats” on page 62 and “New Document” on page 76.

Import Formats

The **import formats** command allows you to import the various formats from another open document to another document or to a book.

Format:

```

Import Formats [DocObject(docvar)] or [BookObject(bookvar)]
    FromDocObject(fromdocvar) formatTypes;
    
```

Table 69: Import Format Command Options

Option Name	Option Description
BookObject	The book object variable for the target book.
DocObject	The Document object variable of the target document. If not specified (and the BookObject option is not specified) the imported formats will go into the currently active document.
FromDocObject (Required)	The source document for the formats.
FormatTypes	This is a list of format types to import. The following table shows the valid values for these types.

Table 70: Import Format Types

Format Type	Meaning
Pgf	Paragraph Catalog Formats
Font	Character Catalog Formats.
Page	Page Layouts
Table	Table Formats
Cond	Condition Formats
RefPage	Reference Pages
Var	Variable Formats.
XRef	Cross-reference Formats
Color	Colors
Math	Math Equation Settings
FBA	<i>FM 10 or greater</i>
All	<i>FM 10 or greater</i>
RemovePageBreaks	Remove all forced page breaks from the target document(s).
RemoveExceptions	Remove exception formats from the target document(s).
CombinedFonts	Import combined fonts
DocumentProps	Import document properties

Example:

The following example imports the paragraph and character formats from a document (identified by the `docvar` object variable), into the currently active document.

```
Open document File('testimportformats.fm') NewVar(docvar);
. . .
Import Formats FromDocObject(docvar) Pgf Font;
```

Import ElementDefs

The **Import ElementDefs** command allows you to import the structural element definitions from another open document or book to another document or book.

Format:

```
Import ElementDefs [DocObject(docvar)] or [BookObject(bookvar)]
  FromDocObject(fromdocvar) ImportFlags;
```

Table 71: Import ElementDefs Options

Option Name	Option Description
BookObject	The book object variable for the target book.
DocObject	The Document object variable of the target document. If not specified (and the BookObject option is not specified) the imported element definitions will go into the currently active document.
FromDocObject (Required)	The source document or book for the element definitions.
ImportFlags	This is a list of options for the import. The following table shows the valid values for these types.

Table 72: Import Flags

Flag	Meaning
RemoveOverrides	Clear format overrides
RemoveBookInfo	Clear formatting inherited from the parent book, if the target is a document
DoNotImportEDD	If the source document is an EDD, just import the element catalog and not the EDD information
NoNotify	Don't trigger a PreImportElemDefs or a PostImportElemDefs event.
DeleteEmptyPages	
DoNotImportStructApp	<i>FM 11 or greater</i>

Example:

The following example imports the element definitions from a document (identified by the docvar object variable), into the currently active document.

```
Open document File('testimportEltDefs.fm') NewVar(docvar);
. . .
Import ElementDefs FromDocObject(docvar);
```

Install Script

The **Install Script** command registers a script with ElmScript. If the script is an event script, the script is loaded into memory and the 'Initialize' Event is run. If it is a standard script, the menu item (under the ElmScript -> Scripts menu) is created as a shortcut to run this script. Note: the user may also install a script via a menu command.

Format:

```
Install Script File(scriptfilename) [Name(scriptname)]
[Label(menulabel)] [Shortcut(string)] When(whenenabled);
```

Table 73: Install Script Options

Option Name	Option Description
File (<i>Required</i>)	A string value containing the name of the script file.
Name	The internal name of the script. If not specified this name will be created by ElmScript. It is used to uninstall the script (if desired) and it is also used to identify the script for message events.
Label	The menu label for standard scripts.
Shortcut	A string value specifying the keyboard shortcut. The text for the shortcut has the following forms: For Alt-Fn (where Fn is any function key) use ' <code>~/Fn></code> ' For Ctl-n (where n is any letter) use ' <code>^n</code> ' For Ctl-Fn (where Fn is any function key) use ' <code>^/Fn></code> ' For Sft-Fn (where Fn is any function key) use ' <code>+/Fn></code> ' For the standard (platform independent) Frame shortcuts use ' <code>\!mn</code> ', where mn is a series of letters. This shortcut is activated by typing in ESC mn, where m and n are the two letters chosen.
When or EnabledWhen	This value determines when the menu command is available. Select from one of the values in the enabled-when table, see "EnabledWhen Context Descriptions" on page 65. If omitted, then the default value will be used, which means it will always be enabled.

The following table lists the values `EnabledWhen` can have and the corresponding contexts in which a menu item is active

Table 74: EnabledWhen Context Descriptions (Page 1 of 2)

EnabledWhen value	Command Context Description
<code>EnableAlwaysDisable</code>	No context. The menu item is disabled. If a menu item is enabled and you set <code>EnabledWhen</code> to this value, it disables and dims the menu item.
<code>EnableAlwaysEnable</code>	All contexts. This is the default value. If the menu item is disabled, setting <code>EnabledWhen</code> to this value enables it.
<code>EnableCanPaste</code>	The Clipboard contains an object or text that can be pasted at the insertion point.
<code>EnableCopy</code>	Some text or an object is selected.
<code>EnableCopyFont</code>	The insertion point or selection is in the text of a paragraph, a math object, a table, or a text line.
<code>EnableInCellText</code>	The insertion point or selection is in a table cell.
<code>EnableInFlow</code>	A text frame is selected, or the insertion point or selection is in a paragraph.
<code>EnableInMath</code>	The insertion point or selection is in a math object.
<code>EnableInParaText</code>	The insertion point or selection is in a paragraph (but not in a math object).
<code>EnableInTable</code>	The insertion point or selection is in any part of a table.
<code>EnableInTableTitle</code>	The insertion point or selection is in the table title.
<code>EnableInText</code>	The insertion point or selection is in a graphic text line or a paragraph.
<code>EnableInTextLine</code>	The insertion point or selection is in a graphic text line.
<code>EnableIsAFrame</code>	The first selected object is an anchored frame.
<code>EnableIsCell</code>	A single cell in a table is selected.

Table 74: EnabledWhen Context Descriptions (Page 2 of 2)

EnabledWhen value	Command Context Description
EnableIsCells	One or more cells in a table are selected.
EnableIsGraphicInset	The first selected object is a graphic inset.
EnableIsObj	An object is selected.
EnableIsOrInFrame	The selected object is a graphic frame or is in a graphic frame that is not a page frame.
EnableIsTable	An entire table is selected.
EnableIsTextFrame	A text frame is selected.
EnableIsTextInset	The first selected object is a text inset.
EnableIsTextSel	The selection is in a paragraph.
EnableIsViewOnly	The current document is locked.
EnableNeedsBookpOnly	A book is open.
EnableNeedsDocpOnly	A document is open.
EnableObjProps	The insertion point is in text, a table, or a math object, or a graphic object is selected.
EnableNeedsDocPOrBookP	A document or a book is open.
EnableBookHasSelection	The book has a selection made.
EnableDocOrBookHasSelection	A document is in front or a book has a selection.

Example 1:

The example installs the event script called `tabletest.fsl`.

```
Install Script File('tabletest.fsl') Name('TableTest');
```

Example 2:

The example installs the standard script called `insertvar.fsl` and supplies a label for the Scripts menu with the keyboard shortcut (ESC a b). The menu will be enabled only when the insertion point is in a text object (paragraph or text line).

```
Install Script File('insertvar.fsl') Name('InsertVariable')
Label('Insert a Variable Here') ShortCut('\!ab') When(EnableInText);
```

Leave

LeaveLoop

The **LeaveLoop** command jumps out of the current loop and starts the script again after the **EndLoop** command.

Format:

```
LeaveLoop;
```

Example:

This example searches the list of paragraph formats in the currently active document for the first one which uses autonumbering. If it finds one, it saves the paragraph format object into the `gvFoundPgFormat` variable and does the `LeaveLoop` command to skip the check all the other paragraph formats in the list.

```

Set gvFoundPgFormat = NULL;
Loop ForEach(PgfFmt) In(ActiveDoc) LoopVar(gvPgFormat)
  If gvPgFormat.PgfIsAutoNum
    Set gvFoundPgFormat = gvPgFormat;
    LeaveLoop;
  EndIf
EndLoop
If gvFoundPgFormat
  Display 'First format with auto numbering is '+gvFoundPgFormat.Name;
Else
  Display 'No paragraph formats with auto numbering exist in this doc';
EndIf

```

LeaveScript

The `LeaveScript` command quits the current script.

Format:

```
LeaveScript;
```

Example:

This example checks to see if there is an active document. If not, it displays a message and the script will stop running.

```

If ActiveDoc = 0
  MsgBox "No Active Document";
  LeaveScript
EndIf
. . .

```

LeaveSub

The `LeaveSub` command jumps out of the current subroutine and continues running as if the subroutine ended normally.

Format:

```
LeaveSub;
```

Example:

This example searches the list of paragraph formats in the currently active document for the first one which uses autonumbering.

```

Sub LookUpPgffFormatAutonum returns pvReturnPgff
  Local lvPgffFormat;
  Loop ForEach(PgffFmt) In(ActiveDoc) LoopVar(lvPgffFormat)
    If lvPgffFormat.PgffIsAutoNum
      Set pvReturnPgff = lvPgffFormat;
      LeaveSub;
    EndIf
  EndLoop
EndSub

```

Example:

The following script illustrates a useful ElmScript subroutine. This subroutine checks to see if a document is already open in the current FrameMaker session. The double slash indicates a comment entry. Any text after the double slash will not be processed and is only there to document the script for the scriptwriter.

```

. . .
//-----
// This subroutine checks the list of open documents to see if the
// specified document is already open.
//
// Format:
// Run IsDocAlreadyOpen pvFileName(testfilename) returns pvDocVar(retdocvar)
//
// If it is open, it returns the document object if it is already open
// and returns zero if it is not open.
//-----
Sub IsDocAlreadyOpen using pvFileName pvDocVar

  // Upper case the string
  Local lvTestDocVar(0);
  Set pvDocVar = 0;
  Loop foreach(Doc) In(Session) LoopVar(lvTestDocVar)
    If eStr.Equal{lvTestDocVar.Name, pvFileName, 'NoCase'}
      Set pvDocVar = lvTestDocVar;
      LeaveLoop;
    EndIf
  EndLoop
EndSub
//-----

```

Local Command

The Local command declares the named variable to be local to the current subroutine. This is useful for writing subroutines and keeping the information in the subroutine separate from the rest of a script. Ordinarily all variables created in a script are global (accessible) to the entire script. Therefore, in a subroutine, you would have to worry about possible naming conflicts with other subroutines or with the main part of the script. This command lets you create variables that are only accessible from within the subroutine.

Format:

```
Local lvVar1[(expression)] [lvVar2[(expression)]];
```

Examples:

This example creates two variables var1 and var2 in the main script and assigns them integer values. It then calls a subroutine which has a local variable called var2. This var2 is different than the var2 from the main part of the script. var1, however, is a global variable and a new version was not declared in the subroutine. So the display from the following script should be;

- Var1=10 Var2=99
- In Sub1--Var1=30 Var2=88
- Var1=30 Var2=99

```
SET var1 = 10;
Set var2 = 99;
Display 'Var1='+var1+' Var2='+var2;
Run Sub1;
Display 'Var1='+var1+' Var2='+var2;

Sub Sub1
  Local var2(0);
  Set var1 = 30;
  Set var2 = 88;
  Display 'In Sub1--Var1='+var1+' Var2='+var2;
EndSub
```

IMPORTANT: The preferred convention is to use lv as a prefix for local variables. The previous example omits this prefix just to illustrate the scope of a local variable. If you used the lv prefix, it would make the script much easier to read.

Loop, EndLoop

The loop command allows to execute a sequence of commands repeatedly until a predefined set of conditions occur. The command has two formats.

Format 1:

```
Loop [While(expression)] or [Until(expression)]
  LoopVar(loopvarname)
  InitVal (initialvalue) Incr(incrementExpression)
  ...
EndLoop
```

Table 75: Loop While/Until Options

Option Name	Option Description
While	The loop continues as long as the expression evaluates to True. It stops when it evaluates to False. For this type of loop you must supply a While or Until option.
Until	The loop continues as long as the expression evaluates to False. It stops when it evaluates to True. For this type of loop you must supply a While or Until option.

Table 75: Loop While/Until Options

Option Name	Option Description
LoopVar (Required)	This option identifies the loop variable. When the loop starts this variable is given the value in the <code>InitVal</code> option and it is incremented by the <code>Incr</code> expression at each subsequent iteration of the loop.
InitVal (Required)	This option specifies the initial value of the loop variable the first time through the loop. When the loop starts, this is equivalent to the following command. <code>Set loopvarname = initialvalue;</code>
Incr (Required)	This option specifies the increment value of the loop variable at each subsequent iteration of the loop. When the loop continues, this is equivalent to the following command. <code>Set loopvarname = loopvarname + incrementExpression;</code>

When this version of the loop command starts, the loop variable (`LoopVar`) is set to the value defined in the `InitVal` option and the while and/or until expressions are tested. If these tests fail (the `While` evaluates to `False` or the `Until` evaluates to `True`), the script continues after the `EndLoop` command. Whenever the script reaches the `EndLoop` command, the Loop variable is incremented by the value in `Incr` and the `While` and/or `Until` conditions are tested again.

Examples:

This example runs through the loop ten times, modifying the variable `gvIndex` with values 1 through 10.

```
Set gvCount = 0;
Loop LoopVar(gvIndex) While(gvIndex <= 10) InitVal(1) Incr(1)
  Set gvCount = gvCount + gvIndex;
EndLoop
write console 'The sum of the numbers between 1 and 10 is '+gvCount;
```

This example runs through the loop ten times, modifying the variable `gvIndex` with values 10 through 1.

```
Set gvCount = 0;
Loop Until(gvIndex < 1) LoopVar(gvIndex) InitVal(10) Incr(-1)
  Set gvCount = gvCount + gvIndex;
EndLoop
write console 'The sum of the numbers between 1 and 10 is '+gvCount;
```

Format 2:

```
Loop ForEach(objectType) In(objectvar) LoopVar(varname)
  commandlist;
EndLoop
```

Table 76: Loop ForEach Options

Option Name	Option Description
ForEach (Required)	This option identifies an object type. The following table gives a list of the allowable object types with the corresponding objects, specified in the <code>In</code> option.

Table 76: Loop ForEach Options

Option Name	Option Description
In	This option identifies an actual object variable. This is used in conjunction with the ForEach option. The following table gives a list of the allowable object types with the corresponding objects, specified in the In option. If not specified, a session object is assumed.
LoopVar	This option identifies the loop variable. When the loop starts this variable is given the value in the InitVal option and it is incremented by the Incr expression at each subsequent iteration of the loop.

This format of the loop command allows you to loop through a one of the FrameMaker predefined lists. It is useful when you wish to do a set of commands on every document in a book, or every document in the current session, or every paragraph in a document (among many others, see the following table).

Table 77: Loop ForEach Types

ForEach Object Type	Source Object Type	Loop Through
AFrame	Sub Column	Each Anchored Frame in the Sub Column.
AFrame	Text Frame	Each Anchored Frame in the TextFrame.
BodyPage	Doc	Each Body Page in the Document.
Book	Session	Each Book in the session
BookComponent	Book	Each Book Component in the Book. For FM9, it follows the hierarchy of component types.
Cell	Sub Column	Each Table cell in the Sub Column.
Cell	Table Row	Each Table cell in the Table row.
Cell	Text Frame	Each Table cell in the Text Frame.
CharFmt	Doc	Each Character Format in the Document.
Color	Doc	Each Color in the Document.
Command	Session	Each Command in the session
CondFmt	Doc	Each Condition Format in the Document.
Doc	Session	Each Document in the session
File	Directory	Each file in a directory on your hard disk.
Flow	Doc	Each Flow in the Document.
Fn	Doc	Each Footnote in the Document.
Fn	Sub Column	Each Footnote in the Sub Column.
Fn	Table	Each Footnote in the table.
Fn	Text Frame	Each Footnote in the Text Frame.
Marker	Doc	Each Marker in the Document
MasterPage	Doc	Each Master Page in the Document.
Member	ListVariable	Each member in a list type variable (e.g. StringList).
Menu	Menu	Each Menu in the another menu.
Menu	Session	Each Menu in the session

Table 77: Loop ForEach Types

ForEach Object Type	Source Object Type	Loop Through
Pgf	Cell	Each Paragraph in the Cell.
Pgf	Doc	Paragraph in the Document
Pgf	Flow	Each Paragraph in the Flow.
Pgf	Footnote	Each Paragraph in the Footnote.
Pgf	Sub Column	Each Paragraph in the Sub Column.
Pgf	Text Frame	Each Paragraph in the TextFrame.
PgfFmt	Doc	Each Paragraph Format in the Document.
RefPage	Doc	Each Reference Page in the Document.
Row	Table	Each Table row in the Table.
RulingFmt	Doc	Each Ruling Format in the Document.
SubCol	Text Frame	Each Sub Column in the Text Frame.
Tbl	Doc	Each Table in the Document.
TblFmt	Doc	Each Table Format in the Document.
TextFrame	Flow	Each Text Frame in the Flow.
Var	Doc	Each Variable in the Document.
VarFmt	Doc	Each Variable Format in the Document
XRef	Doc	Each Cross-reference in the Document
XRefFmt	Doc	Each Cross-reference Format in the Document

Example 1:

Loop through each book component in a book and the name of each document to the FrameMaker console.

```

Loop ForEach (BookComponent) In(ActiveBook) LoopVar(gvBookCompVar)
    Write Console 'Doc name='+gvBookCompVar.Name;
EndLoop

```

Example 2:

Open a document, create a new document and loop through each paragraph of the open document and, if the paragraph format of the paragraph is 'Head1' then write the text of that paragraph to a new paragraph in the new document.

```
Open Document File('testdoc.fm') NewVar(gvInDocVar)
New Document Portrait NewVar(gvOutDocVar);
Loop ForEach (Pgf) In(gvInDocVar) LoopVar(gvPgfVar)
  If gvPgfVar.Name = 'Head1'
    Set gvTFrame = gvOutDocVar.MainFlowInDoc.LastTextFrameInFlow;
    New paragraph NewVar(gvNewPgfVar)
      PrevObject(gvTFrame.LastPgf);
      New Text Object(gvNewPgfVar) gvPgfVar.Text;
    EndIf
  EndLoop
Close Document DocObject(gvInDocVar);
Save Document DocObject(gvOutDocVar) File('testout.fm');
Close Document DocObject (gvOutDocVar);
```

Example 3:

Loop through each file in the specified directory.

```
Loop ForEach (File) In('C:\Program files\ElmScript4\') LoopVar(gvFileName)
  Write Console 'File name='+gvFileName;
EndLoop
```

Example 4:

Loop through each string in the string list variable and writes the string to the console.

```
New StringList NewVar(gvStrList) Value('String1') Value('String2') Value('String3');
Loop ForEach (Member) In(gvStrList) LoopVar(gvStringValue)
  write console 'String = '+gvStringValue;
EndLoop
```

LoopNext

The **LoopNext** command jumps out of the current loop and starts again at the next iteration of the loop.

Format:

```
LoopNext;
```

Example:

This example has a loop with gvIndex going from 1 to 10, but when the gvIndex variable is 7, the rest of the loop is skipped and continues with the 8, 9, and 10 values.

```
Loop LoopVar(gvIndex) While(gvIndex <= 10) InitVal(1) Incr(1)
  If gvIndex = 7 LoopNext EndIf
  . . .
EndLoop
```

Merge

The **Merge** command merges the selected structural element or elements from the specified document into the first or last element in the selection. *Structured FrameMaker only.*

Format:

```
Merge [DocObject (docvar)] [BookObject(bookvar)]
      [IntoFirst] [IntoLast]
      [ElementRange(eltrangeVar)];
```

Table 78: Merge Options

Option Name	Option Description
DocObject	The document object variable from which to merge the elements. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to merge the elements. If not specified and the Doc object is also not specified, the currently active book is used.
IntoFirst	Merge the elements into the first one in the range (Default)
IntoLast	Merge the elements into the last one in the range.
ElementRange	The range of elements to merge. If not specified then the current element selection will be used.

Example

The following script merges the current element selection of the currently active document into the first element in the range.

```
. . .
Merge;
. . .
```

See also

“Demote” on page 27, “Wrap” on page 149, and “Unwrap” on page 144.

MsgBox

The MsgBox command displays a message in a dialog box. There are six different modes for this dialog box, which puts up a set of buttons for the user to push.

Format:

```
MsgBox expression [Mode(modetype)] [Button(buttonpushedvar)];
```

Table 79: MsgBox Options

Option Name	Option Description
expression (Required)	A computational expression usually a string.
Mode	<p>This identifies the type of buttons to put in the box.</p> <p>Possible values:</p> <p>OkCancel Ok and Cancel buttons, Ok is default.</p> <p>CancelOk Ok and Cancel buttons, Cancel is default.</p> <p>YesNo Yes and No buttons, Yes is default.</p> <p>NoYes Yes and No buttons, No is default.</p> <p>Warn Ok button with a warning icon.</p> <p>Note Ok button with a note icon.</p> <p>YesNoCancel Yes, No and Cancel Buttons, Yes is default. <i>Frame 7.0 or greater.</i></p>
Button	<p>This option identifies the variable where ElmScript will store the value from which button the user pushed.</p> <p>Possible Values:</p> <p>OkButton User pressed the Ok button.</p> <p>CancelButton User pressed the Cancel button.</p> <p>YesButton User pressed the Yes button.</p> <p>NoButton User pressed the No button.</p>

Example:

This example displays a message and lets the user select a button.

```

MsgBox 'Press Yes for Yes and No for No' Mode(YesNo)
      Button(gvButtonVar);
If gvButtonVar = YesButton
  MsgBox 'The user pressed the Yes button';
Else
  MsgBox 'The user pressed the No button';
EndIf

```

New Commands

The New command is used to create something in ElmScript. This command can be used to create FrameMaker documents and books, as well as all the objects that they contain, such as paragraphs, markers, etc. It can also be used to add text to a paragraph. In addition to FrameMaker objects, the New command can be used to create ElmScript variables, structures, lists, and EslObjects. The items below describe how to use the New command to create these various objects and data.

New Project (See `EsLObject` reference *FM 2017* or greater)

New Document

The **New Document** command creates a new FrameMaker document. Documents can be created directly, specifying the dimensions of the document page, using a predetermined set of dimensions (Portrait or Landscape), using a template from an existing document, or having the user select a document template.

The New document command not only creates a document but it also allows you to set document properties in the same command. Simply put the properties on the same line as the other options.

Format 1:

```
New Document NewVar(gvDocVar);
```

This format of the **New Document** command displays a dialog box allowing the user to choose a template for the new document

Table 80: New Document Dialog Options

Option Name	Option Description
NewVar	The name of the variable to hold the newly created document object.

Example:

This sample displays a dialog box requesting the user to select a template for the new document. When the template is selected a new document is created, represented in the variable `docvar` and the number of print copies is set to 2.

```
. . .
New Document NewVar(gvDocVar) Printcopies(2);
. . .
```

Format 2:

```
New Document Template(filename) NewVar(gvDocVar) [openoptions];
```

This format of the **New Document** command creates a new FrameMaker document based on the template file specified.

Table 81: New Document from Template Options

Option Name	Option Description
Template	This option specifies the file name of the template.
openoptions	The open options are a variety of options for opening the template. See open options for more detail.
NewVar	The name of the variable to hold the newly created document object.

Example:

This example creates a new document from the template in file `c:\templates\mytemp.tpl` bypassing any messages if the template is from an older version of FrameMaker.

```
. . .
New Document Template ('c:\templates\mytemp.tpl') NewVar(gvDocVar) FileIsOldVersion;
. . .
```

Format 3:

```
New Document {Portrait Landscape} NewVar(gvDocVar);
```

This format of the **New Document** command creates a new FrameMaker document based on predetermined dimensions.

Table 82: New Document from Predetermined Values Options

Option Name	Option Description
Predetermined Type	This option specifies the type of predetermined dimensions. Values are: Portrait Landscape
NewVar	The name of the variable to hold the newly created document object.

Table 83: Predetermined Values for New Document Creation

Document parameter	Portrait	Landscape
Document Width	8.5 inches	11 Inches
Document Height	11 Inches	8.5 Inches
Column Gap size	.25 Inches	.25 Inches
Top Margin	1 Inch	1 Inch
Bottom Margin	1 Inch	1 Inch
Left Margin	1 Inch	1 Inch
Right Margin	1 Inch	1 Inch
Number of Columns	1	1
SidedNess	SingleSided	SingleSided
Visible	True	True

Example:

This sample creates a new portrait document.

```
. . .
New Document Portrait NewVar(gvDocVar);
. . .
```

Format 4:

```

New Document NewVar(gvDocVar);
  Width(gvDocWidth) Height(gvDocHeight)
  NumCols(gvNumberOfColumns) ColumnGap(gvGapWidth)
  TopMargin(gvTopMargin) BottomMargin(gvBottomMargin)
  LeftInsideMargin(gvLeftInsideMargin) RightOutsideMargin(gvRightOutsideMargin)
  {SingleSided FirstPageRight FirstPageLeft}
  [Invisible]

```

This format of the **New Document** command creates a new FrameMaker document based on the specified dimensions. The default values for all these options is the same as the portrait predetermined type. You need only specify the dimensions that are different from the portrait.

Table 84: New Document from Specified Values Options

Option Name	Option Description
Width	The width of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
Height	The height of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
NumCols	The number of columns
ColumnGap	The size of the gap between columns in the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
TopMargin	The top margin of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
BottomMargin	The Bottom margin of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
LeftInsideMargin	The left inside margin of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
RightOutsideMargin	The right outside margin of the document. This is a metric measurement. Make sure to specify the units, such as Inches. The default units type is points (1/72 of an inch).
SidedNess	This specifies how master pages are automatically applied. The possible values are: SingleSided FirstPageRight FirstPageLeft
Invisible	If specified the document will be created invisible; otherwise visible.
NewVar	The name of the variable to hold the newly created document object.

Example:

This example creates a new document 8 inches wide, 7.5 inches high with 2 columns, double sided with the first page as the right page.

```

. . .
New Document NewVar(gvDocVar) width(8") height(7.5") NumCols(2) FirstPageRight;
. . .

```

See also

“Open Document or Book” on page 112, “Close Document” on page 12, “Save Document or Book” on page 132.

New FrameMaker Objects

This variation of the New command creates new FrameMaker objects. When a new object is created, you can set the properties for that object on the same command line. The commands can be divided into six formats.

The New object types commands not only creates the FrameMaker objects but it also allows you to set object properties in the same command. Simply put the properties on the same line as the other options.

New Anchored Objects

New AFrame
New Footnote
New Marker
New TiApiClient
New InlineComponent *FM 2015 or greater*

Format:

```
New {AFrame Footnote Marker TiApiClient InlineComponent FM 2015 or greater}
  NewVar(varname)
  [MarkerName(mnamestr)]
  [PgFmtName(pgfFormatName)] (for InlineComponent only)
  [PgFmtNameList(stringlist)] (for InlineComponent only)
  [InlineComponentType(MiniToc)] (for InlineComponent only)
  [AddHyperLinks(true/false)] (for InlineComponent only)
  [DocObject(docvar) or TextLoc(textlocation)];
```

This format of this new command creates anchored FrameMaker objects. Anchored objects are those which occur in text locations.

Table 85: New Anchored Objects Options

Option Name	Option Description
Formatted Type (<i>Required</i>)	The type of anchored object to create. This must be one of the following: AFrame Footnote Marker TiApiClient
TextLoc	The text location variable identifying the place to insert the object. If not specified the insertion point of the document will be used.
MarkerName	If you are creating a new marker, you can optionally identify the marker type name with this option. If not specified, you can change it later using the MarkerTypeId property of the marker.
PgFmtName	If you are creating a new InlineComponent, use this option (use as many as needed) to specify the paragraph tags to use for the minitoc. <i>FM 2015 or greater</i>
PgFmtNameList	If you are creating a new InlineComponent, use this option (use as many as needed) to specify a list (StringList) paragraph tags to use for the minitoc. <i>FM 2015 or greater</i>
AddHyperLinks	If you are creating a new InlineComponent, you can optionally indicate whether to add hyper links. The default value is True. <i>FM 2015 or greater</i>
InlineComponentType	If you are creating a new InlineComponent, you can optionally identify the type of the component with this option. At present, only MiniToc is supported, so this is the default value. <i>FM 2015 or greater</i>

Table 85: New Anchored Objects Options

Option Name	Option Description
DocObject	The document object variable identifying the document to put the new formatted object. If not specified the currently active document will be used.
NewVar (Required)	The name of the variable to hold the newly created object.

Example:

This example creates a new footnote at the insertion point of the currently active document.

```
New Footnote NewVar(vFnVar);
```

See also

“Delete Object” on page 23.

New Anchored Formatted Objects

New Variable**New XRef****Format:**

```
New {Variable XRef}
    NewVar(varname) Format(formatname)
    [DocObject(docvar) or TextLoc(textlocation)];
```

This format of this new command creates anchored formatted FrameMaker objects. Anchored formatted objects are those which occur in text locations and require a format.

Table 86: New Anchored Formatted Objects Options

Option Name	Option Description
Formatted Type (Required)	The type of formatted object to create. This must be one of the following: Variable XRef
Format (Required)	The format of the object to create (a variable format or cross-reference format name).
TextLoc	The text location variable identifying the place to insert the object. If not specified the insertion point of the document will be used.
DocObject	The document object variable identifying the document to put the new formatted object. If not specified the currently active document will be used.
NewVar (Required)	The name of the variable to hold the newly created object.

Example:

This example creates a new system variable with the current page number at the insertion point of the currently active document.

```
New Variable Format('Current Page #')
```

New Format Rule Objects

New FormatChangeList

Format 1:

```
New FormatChangeList
  Name(objectname) NewVar(varname)
  [DocObject(docvar)];
```

Format 2:

```
New FormatChangeList
  ParentObject(parentObj) NewVar(varname);
```

Table 87: New FormatChangeList Object Options

Option Name	Option Description
Name (Required)	The name of the new object.
DocObject	The document object variable identifying the document to put the new object. If not specified the currently active document will be used.
ParentObject	This identifies the frame object which will contain the new FmtChangeList Object. The parent must be FmtRuleClause
NewVar (Required)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new FmtChangeList format in the currently active document.

```
New FormatChangeList Name('MyNewChangeList')
```

Example 2:

This example creates a new FmtChangeList format associated with a FmtRuleClause.

```
Get Object Name('MyElementDef') Type(ElementDef) NewVar(eltDefVar);
New FormatRule ParentObject(eltDefVar) RuleType(PrefixRules) NewVar(ruleVar)
New FormatRuleClause ParentObject(ruleVar) NewVar(ruleClauseVar);
New FormatChangeList ParentObject(ruleClauseVar);
```

See also

“New Structured Objects” on page 90 and “New FmtRuleClause” on page 82.

New FormatRule

Format:

```
New FormatRule
  [RuleType(type)]
  ParentObject(parentobj) NewVar(varname);
```

This command allows you to associate the new FmtRule object with a specified property of the parent object. For example, you can create an FmtRule as the text format rule of an element definition or as a subformat rule of a format rule clause.

Table 88: New FormatRule Object Options

Option Name	Option Description
ParentObject (<i>Required</i>)	This identifies the frame object which will contain the new object. The parent can be either a <code>FmtRuleClause</code> or an <code>ElementDef</code> .
RuleType	This identifies the type of rule for the new <code>FmtRule</code> object if the parent object is an <code>ElementDef</code> . It is not necessary if the parent is a <code>FmtRuleClause</code> . The default rule type if not specified is <code>TextFmtRules</code> . Possible values are: <code>TextFmtRules</code> - More than one allowed. <code>ObjectFmtRules</code> - Only one allowed. <code>PrefixRules</code> - More than one allowed. <code>SuffixRules</code> - More than one allowed. <code>FirstPgfRules</code> - More than one allowed. <code>LastPgfRules</code> - More than one allowed.
NewVar (<i>Required</i>)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new `FmtRule` object for the element definition called `MyElementDef`. The rule type is `PrefixRules`.

```
Get Object Name('MyElementDef') Type(ElementDef) NewVar(eltDefVar);
New FormatRule ParentObject(eltDefVar) RuleType(PrefixRules);
```

See also

“New Structured Objects” on page 90.

New FmtRuleClause**Format:**

```
New FormatRuleClause
    ParentObject(parentobj) NewVar(varname);
```

This command allows you to create a new `FmtRuleObject` object associated with a `FmtRule` parent object.

Table 89: New FormatRuleClause Object Options

Option Name	Option Description
ParentObject (<i>Required</i>)	This identifies the <code>FmtRule</code> object.
NewVar (<i>Required</i>)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new `FmtRuleClause` Object for a recently created `FmtRule` Object.

```
Get Object Name('MyElementDef') Type(ElementDef) NewVar(eltDefVar);
New FormatRule ParentObject(eltDefVar) RuleType(PrefixRules) NewVar(ruleVar)
New FormatRuleClause ParentObject(ruleVar);
```

See also

“New Structured Objects” on page 90.

New Graphic Objects

- New Arc
- New Ellipse
- New Flow
- New Group
- New Inset
- New Line
- New Math
- New MathML (*FM 12 or greater*)
- New Polygon
- New Polyline
- New Rectangle
- New RoundRect
- New TextFrame
- New Textline
- New UnanchoredFrame

Format:

```
New {Arc Ellipse Flow Group Inset Line Math Polygon Polyline
    MathML (FM 12 or greater)
    Rectangle RoundRect TextFrame TextLine UnanchoredFrame}
    ParentObject(parentobj) NewVar(varname);
```

This format of the new command creates graphic type objects.

Table 90: New Graphic Object Options

Option Name	Option Description
GraphicType (<i>Required</i>)	The type of graphic to create. This must be one of the following: Arc Ellipse Flow Group Inset Line Math MathML PolyLine Polygon Rectangle RoundRect TextFrame TextLine UnanchoredFrame
ParentObject (<i>Required</i>)	This identifies the frame object which will contain the new object.
NewVar (<i>Required</i>)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new Text frame on the first body page of the currently active document. The new text frame will be 5 inches wide and 5 inches high.

```
Set pageObj = FirstBodyPageInDoc; // Get first body page
Set FrameObj = pageObj.PageFrame; // Get page frame
// This next command creates a new Text Frame using the
// above page frame as the parent. The new Text Frame will
// five inches wide and five inches in height.
New TextFrame NewVar(newFrameObj) ParentObject(FrameObj)
Width(5") Height(5");
```

Example 2:

This example creates a new anchored frame at the insertion point of the currently active document and then creates an arc and a rectangle in that frame, setting their properties at the time of creation.

```
New AFrame NewVar(afrm) Height(5.5") Width(4.75")
AnchorType(AnchorRunIntoParagraph)
New Arc ParentObject(afrm) width(2") LocX(0.5")
New Rectangle ParentObject(afrm) width(1") height(1");
```

See also

“Delete Object” on page 23.

New Menu Commands

New Command

Format:

```
New Command
[Name(objectname)] Label(stringexpression) [NewVar(varname)]
EventProc(eventname) [Shortcut(stringExpression)
[AddTo(menuvar) ];
```

The **New Command** command creates menu item command in the current FrameMaker session.

Table 91: New Command Options

Option Name	Option Description
Label (<i>Required</i>)	The text of the label that will appear on the menu for the command object.
Name	The name of the new command. If not provided, a random name will be assigned.
EventProc (<i>Required</i>)	The name of the event that will be run whenever the user clicks on this menu item. Somewhere in your script there should be an event command followed by the name that you put in this option.
NewVar	The name of the variable to hold the newly created object.
Shortcut	The text of the keyboard shortcut to invoke this command.
AddTo	This is a menu object to which the command will be added. Alternatively you may omit this option and use the Add CommandObject command later to add it to a menu.

Examples:

This example creates a new menu and new command and puts it an event to display a message when the user clicks on the command.

```

. . .
New Menu Label('My First Menu') NewVar(mymenuobj) Addto('!MakerMainMenu');
New Command Name('MyFirstCommand') Addto(mymenuobj) EventProc(myFirstMenuEvent)
    Label('My First Menu Command');
. . .

Event myFirstMenuEvent
    MsgBox 'The user pressed my menu command';
EndEvent

```

See also

“Add CommandObject” on page 1 and “Remove CommandObject” on page 125.

New Menu**Format:**

```

New Menu
    [Name(objectname)] Label(stringexpression) [NewVar(varname)]
    [AddTo(menuvar)];

```

This command creates menu item command in the current FrameMaker session.

Table 92: New Menu Options

Option Name	Option Description
Label (<i>Required</i>)	The text of the label that will appear on the menu.
Name	The name of the new menu created. If not provided, a random name will be assigned.
NewVar	The name of the variable to hold the newly created menu.
AddTo	This is a menu or menubar object to which the command will be added. Alternatively you may omit this option and use the Add MenuObject command later to add it to a menu or menubar.

Table 93: FrameMaker Menubar Names

Menubar names	Menubar Description ^a
!MakerMainMenu	The name of the FrameMaker main menu bar.
!ViewOnlyMainMenu	The name of the FrameMaker view only menu.
!QuickMakerMainMenu	The name of the FrameMaker quick menu.
!BookMainMenu	This name of the FrameMaker Book menu.
!ViewOnlyBookMainMenu	This name of the FrameMaker View only Book menu.
!QuickBookMainMenu	This name of the FrameMaker Quick Book menu.

a. For a complete list of predefined FrameMaker menubar, menu and command names, see the file 'Menus.cfg' in the **fminit\maker** subdirectory located in the **maker** directory.

Table 94: FrameMaker Menu Names

Menu names	Menu Description ^a
FileMenu	The name of the FrameMaker main menu bar's File menu.
EditMenu	The name of the FrameMaker main menu bar's Edit menu.
FormatMenu	The name of the FrameMaker main menu bar's Format menu.
BookFileMenu	This name of the FrameMaker book menu bar's File menu.

a. For a complete list of predefined FrameMaker menubar, menu and command names, see the file 'Menus.cfg' in the **fminit\maker** subdirectory located in the **maker** directory.

Examples:

This example creates a new menu and new command and puts in an event to display a message when the user clicks on the command.

```

. . .
New Menu Label('My First Menu') NewVar(mymenuobj) Addto('!MakerMainMenu');
New Command name('MyFirstCommand') Addto(mymenuobj) EventProc(myFirstMenuEvent)
  label('My First Menu Command');
. . .

Event myFirstMenuEvent
  MsgBox 'The user pressed my menu command';
EndEvent

```

See also

“Add MenuObject” on page 3 and “Remove MenuObject” on page 126.

New MenuItemSeparator**Format:**

```

New MenuItemSeparator
  [Name(objectname)] [NewVar(varname)] [AddTo(menuvar)];

```

This command creates menu item separator in the current FrameMaker session.

Table 95: New Menu Item Separator Options

Option Name	Option Description
Name	The name of the new menu item separator. If not provided, a random name will be assigned.
NewVar	The name of the variable to hold the newly created object.
AddTo	This is a menu object to which the menu item separator will be added. Alternatively you may omit this option and use the Add CommandObject command later to add it to a menu.

Examples:

This example creates a new menu item separator and a new command, puts them both under the FrameMaker 'Edit' menu, and puts in an event to display a message when the user clicks on the command.

```

. . .
Get Object Type(Menu) Name('EditMenu') NewVar(frameeditmenuobj);
New MenuItemSeparator Name('MySeparator') Addto (frameeditmenuobj);
New Command name('MyFirstCommand') Addto(frameeditmenuobj) EventProc(myFirstMenuEvent)
    label('My First Menu Command');
. . .

Event myFirstMenuEvent
    MsgBox 'The user pressed my menu command under the FrameMaker Edit menu';
EndEvent

```

See also

“Add MenuSepObject” on page 3 and “Remove MenuSepObject” on page 127.

New Named Objects

New Book
New CharacterFormat
New Color
New ConditionFormat
New ElementDef (*Structured FrameMaker only.*)
New FormatChangeList
New MarkerType
New MasterPage
New ParagraphFormat or **PgfFmt**
New ReferencePage
New RulingFormat
New TableFormat
New VariableFormat
New XRefFormat
New AttrCondExpr *FM 8.0 or greater*
New GraphicsFmt *FM 11 or greater*
Format:

```

New {Book CharacterFormat Color ConditionFormat ElementDef MarkerType
    MasterPage ParagraphFormat ReferencePage RulingFormat GraphicsFmt
    TableFormat VariableFormat XRefFormat FormatChangeList AttrCondExpr}
Name(objectname) NewVar(varname)
[DocObject(docvar)];

```

This format of this new command creates named FrameMaker objects.

Table 96: New Named Objects Options

Option Name	Option Description
Named Type (<i>Required</i>)	The type of named object to create. This must be one of the following: Book CharacterFormat Color ConditionFormat MarkerType MasterPage ParagraphFormat ReferencePage RulingFormat TableFormat VariableFormat XRefFormat FormatChangeList AttrCondExpr GraphicsFmt
Name (<i>Required</i>)	The name of the new object. For a Book the name is the file name; otherwise it is a FrameMaker internal name.

Table 96: New Named Objects Options

Option Name	Option Description
DocObject	The document object variable identifying the document to put the new named object. If not specified the currently active document will be used.
NewVar (Required)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new paragraph format in the currently active document.

```
New ParagraphFormat name('MyNewPgfFmtName')
```

New Series Objects

New BodyPage

Format 1:

```
New BodyPage PrevObject(parentobj)
  [DocObject(docvar)] [NewVar(varname)];
```

Table 97: New BodyPage Options

Option Name	Option Description
DocObject	For Pgf and BodyPage types: The document object variable identifying the document in which to put the new series object. If not specified and the PrevObject option is not specified, the currently active document will be used.
PrevObject	This identifies the frame object which will come before the new object in the list. To add a paragraph at the start of a flow, use the flow object here. To insert it at the beginning of the other lists, leave this option off.
NewVar (Required)	The name of the variable to hold the newly created BodyPage object.

Example:

This example creates a new immediately after the current page.

```
Set pageObj = ActiveDoc.CurrentPage; // Get first body page
New BodyPage PrevObject(pageObj) NewVar(newPageObj);
```

New BookComponent

Format 1:

```
New BookComponent PrevObject(parentobj)
  [DocObject(docvar)] [NewVar(varname)];
```

This format of the new BookComponent command creates a book component after a specified parent.

Format 2:

```

New BookComponent Name(compnamevar)
  ElementLoc(elementlocvar) [BookObject(bookvar)]
  [CompType(cmpType)] FM 9.0 or greater
  [NewVar(varname)];

```

This format of the new BookComponent command creates a new book component at the specified position in a structured book. The location is given by the ElementLoc option. *Structured FrameMaker only.*

Table 98: New BookComponent Object Options

Option Name	Option Description
BookObject	For the BookComponent type: The book object variable identifying the book in which to put the new book component object. If not specified and the ElementLoc option is not specified, the currently active book will be used.
PrevObject	This identifies the frame object which will come before the new object in the list. To add a paragraph at the start of a flow, use the flow object here. To insert it at the beginning of the other lists, leave this option off.
Name	A string indicating the name of the book component to add.
ElementLoc	The element location for adding the new component
CompType	The type of the component to create (BkFolder or BKGroup). <i>FM 9.0 or greater</i>
NewVar (Required)	The name of the variable to hold the newly created object.

Example 1:

This example creates a new Book component in the currently active book and sets the file name as 'c:\MyDir\MyFile.fm'.

```

Set comp1 = ActiveBook.FirstComponentInBook;
New BookComponent PrevObject(comp1) NewVar(bkCompVar);
Set bkCompVar.Name = 'c:\MyDir\MyFile.fm';

```

Example 2:

This example inserts a new book component into the structured hierarchy.

```

New ElementLoc NewVar(elocvar) Parent(0)
  Child(ActiveBook.HighestLevelElement) offset (0);
New BookComponent NewVar(ncompvar) Name('Introduction')
  ElementLoc(elocvar);

```

See also

“Delete Object” on page 23.

New Pgf**Format 1:**

```

New Paragraph
  [DocObject(docvar)]
  [PgfFmtName(fmtName)]
  [Text(textData)]
  PrevObject(parentobj) [NewVar(varname)];

```

This command creates a new paragraph object.

Table 99: New Paragraph Object Options

Option Name	Option Description
DocObject	For <code>Pgf</code> and <code>BodyPage</code> types: The document object variable identifying the document in which to put the new series object. If not specified and the <code>PrevObject</code> option is not specified, the currently active document will be used.
PrevObject	This identifies the frame object which will come before the new object in the list. To add a paragraph at the start of a flow, use the flow object here. To insert it at the beginning of the other lists, leave this option off.
PgfFmtName	If the object added is a paragraph, you may optionally use this to specify the paragraph format for the new paragraph. You can set this later using the <code>.Properties</code> property.
Text	If the object added is a paragraph, you may optionally use this to some text for the new paragraph.
NewVar (Required)	The name of the variable to hold the newly created object.

Example:

This example creates a new `Text` frame on the first body page of the currently active document. The new text frame will be 5 inches wide and 5 inches high. Add two paragraphs to the textframe, the first will be tagged with the 'Body' paragraph format. Puts some text in the second one.

```
Set pageObj = FirstBodyPageInDoc; // Get first body page
Set FrameObj = pageObj.PageFrame; // Get page frame
// This next command creates a new Text Frame using the
// above page frame as the parent. The new Text Frame will
// five inches wide and five inches in height.
New TextFrame NewVar(newFrameObj) ParentObject(FrameObj)
Width(5") Height(5");
New Paragraph PrevObject(newFrameObj.Flow) PgfFmtName('Body') NewVar(pgf1Obj);
New Paragraph PrevObject(pgf1Obj) NewVar(pgf2Obj) Text('Add some text right here');
```

See also

“Delete Object” on page 23.

New Structured Objects

New ElementDef

Format:

```
New ElementDef
Name(objectname) NewVar(varname)
[DocObject(docvar)];
```

This format of this new command creates named `FrameMaker` objects. This is for *Structured FrameMaker* only..

Table 100: New Named Objects Options

Option Name	Option Description
Name (Required)	The name of the new Element Definition.
DocObject	The document object variable identifying the document to put the new element definition. If not specified the currently active document will be used.
NewVar (Required)	The name of the variable to hold the newly created element definition.

Example 1:

This example creates a new paragraph format in the currently active document.

```
New ElementDef name('MyElementDef')
```

New Element**Format:**

```
New Element [BookObject(bookvar)] [DocObject(docvar)] NewVar(varname)
  {ElementDef(eltdefvar) ElementDefName(eltvarname)}
  {ElementLoc(eltlocvar) TextLoc(textlocvar)};
```

This format of the new command creates a new structured element at the specified position in a structured doc or book. This is for *Structured FrameMaker* only..

Table 101: New Element Object Options

Option Name	Option Description
DocObject	The document object variable identifying the document in which to put the new element object. If not specified and the ElementLoc, TextLoc and ElementDef options are not specified, the currently active document will be used.
BookObject	The book object variable identifying the book in which to put the new element object. If not specified and the ElementLoc option is not specified, ElmScript will use the DocObject criteria above.
ElementDef	This identifies the object variable of an element definition (ElementDef). Use the Get Object command to get the correct object variable.
ElementDefName	A string indicating the name of the element definition. You can use this in place of the ElementDef option above.
ElementLoc	The element location for adding the new component. If not specified, the TextLoc option (below) will be used.
TextLoc	The text location variable identifying the place to insert the element. If not specified (and the ElementLoc option (above) is not specified) the insertion point of the document will be used.
NewVar (Required)	The name of the variable to hold the newly created object.

Example:

This example creates a new element at the current text location using the Emph element definition.

```
New Element NewVar(eltlvar) ElementDefName('Emph');
```

Example:

This example inserts a new book component into the structured hierarchy.

```
SET eltlocvar = ActiveDoc.ElementSelection.Begin;
New Element NewVar(eltvar) ElementDefName('Para')
    ElementLoc(eltlocvar);
```

New Attribute**Format:**

```
New Attribute NewVar(varname)
    AttrName(name) [AllowSpecial(spFlag)] [Value(objectvar2)];
```

The **New Attribute** command creates an attribute variable. *Structured FrameMaker only.*

Table 102: New Attribute Options

Option Name	Option Description
AttrName	This option specifies the name of the attribute and is required.
AllowSpecial	The option is a flag that is True or False. True, if the attribute is allowed as a special case and False otherwise. The default value is False.
Value	The value of the attribute. This may be repeated many times, one for each value of the attribute. Commonly, one value is used.
NewVar	The name of the variable to hold the newly created attribute.

Example 1:

This sample script creates an attribute variable with one value.

```
. . .
New Attribute NewVar(attrVar) AttrName('MyAttr') AllowSpecial(False)
    Value('MyVal');
. . .
```

Example 2:

This sample script creates an attribute variable with more than one value.

```
. . .
New Attribute NewVar(attrVar) AttrName('MyAttr') AllowSpecial(False)
    Value('MyVal') Value('MyOtherVal');
. . .
```

See also

“Get Object” on page 46.

New AttributeDef

Format:

```

New AttributeDef NewVar(varname)
  AttrDefName(name) [AttrType(type)]
    [Choice(choiceValue)] [Choice(choiceValue)] [...]
    [Default(defaultValue)] [Default(defaultValue)] [...]
    [Required(reqValue)] [Flag(flagValue)]
    [RangeMin(rangeMinValue)] [RangeMax(rangeMaxValue)]

```

The `New AttributeDef` command creates an attribute definition variable. *Structured FrameMaker only.*

Table 103: New AttributeDef Options

Option Name	Option Description
AttrDefName	This option specifies the name of the attribute definition and is required.
AttrType	<p>The option specifies the type of the attribute definition. It indicates the type of data that is allowed for the attribute. The following values are allowed:</p> <ul style="list-style-type: none"> <code>AtString</code> - Any String value. <code>AtStrings</code> - One or more string values. <code>AtChoices</code> - A value from a list of choices. <code>AtInteger</code> - An integer value. <code>AtIntegers</code> - One or more integer values. <code>AtReal</code> - A real value <code>AtReals</code> - One or more real values. <code>AtUniqueID</code> - A string that uniquely identifies the element. <code>AtUniqueIDRef</code> - A reference to a UniqueID Attribute <code>AtUniqueIDRefs</code> - One or more references to a UniqueID attribute. <p>The default value is <code>AtString</code>.</p>
Choice	<p>The list of choices if the attribute type is <code>AtChoices</code>. Repeat the option for each choice allowed.</p> <p>The default value is no choices.</p>
Default	<p>The lists of default values if the attribute is not required. This may be repeated many times, one for each default value of the attribute, if the attribute type contains multiple values.</p> <p>The default value is no defaults.</p>
Required	True if the attribute is required and <code>False</code> otherwise. The default value is <code>False</code> .
Flag	<p>The flag is a bitwise property that can have zero or more of the following values:</p> <ul style="list-style-type: none"> <code>AfReadOnly</code> - The attribute value is read-only. <code>AfHidden</code> - The attribute value is hidden. <p>The default value is none of the above.</p>
RangeMin	If specified, it represents the minimum allowable value that the attribute may take. The default value is <code>No Minimum</code> value.
RangeMax	If specified, it represents the maximum allowable value that the attribute may take. The default value is <code>No Maximum</code> value.
NewVar	The name of the variable to hold the newly created attribute.

Example 1:

This sample script creates an attribute definition variable allowing a string value. It is not required, has no minimum or maximum values.

```
. . .
New AttributeDef NewVar(attrDefVar) AttrDefName('MyAttrDef');
. . .
```

Example 2:

This sample script creates an attribute definition variable that gives the corresponding attribute a list of valid value choices and is required.

```
. . .
New AttributeDef NewVar(attrDefVar) AttrDefName('MyAttrDef') AttrType(AtChoices)
  Choice('Choice1') Choice('Choice2') Choice('Choice3')
  Required(True);
. . .
```

Example 3:

This sample script creates an attribute definition variable that the attribute must have an integer value between 100 and 200, if the attribute is present.

```
. . .
New AttributeDef NewVar(attrDefVar) AttrDefName('MyAttrDef') AttrType(AtInteger)
  RangeMin(100) RangeMax(200);
. . .
```

Example 4:

This sample script creates an attribute definition variable that the attribute allows a string value and if the value is not specified, the value 'MyVal' will be used.

```
. . .
New AttributeDef NewVar(attrDefVar) AttrDefName('MyAttrDef') AttrType(AtString)
  Default('MyVal');
. . .
```

See also

“New ElementDef (Structured FrameMaker only.)” on page 87.

New AttributeList**Format:**

```
New AttributeList NewVar(varname)
  AttrName(name1) [AllowSpecial(spFlag)] [Value(objectvar2)]
  AttrName(name2) [AllowSpecial(spFlag)] [Value(objectvar2)]
  . . .;
```

The **New AttributeList** command creates a list of attributes. Each attribute is defined the same way as for an individual attribute. You may repeat the entry for each attribute in the list. See “New Attribute” on page 92. *Structured FrameMaker only.*

Example:

This sample script creates a attributeList variable with one value.

```
. . .
New AttributeList NewVar(attrListVar)
  AttrName('MyAttr') AllowSpecial(False) Value('MyVal');
. . .
```


Example:

This sample script creates a `attributeList` variable with three values.

```

. . .
New AttributeList NewVar(attrListVar)
  AttrName('MyAttr1') AllowSpecial(False) Value('MyVal')
  AttrName('MyAttr2') Value('MyVal') Value('MyValue2')
  AttrName('MyAttr3') AllowSpecial(True) Value('MyVal-Three');
. . .

```

New AttributeDefList**Format:**

```

New AttributeDefList NewVar(attrDefListVar)
  AttrDefName(name) [AttrType(type)]
    [Choice(choiceValue)] [Choice(choiceValue)] [...]
    [Default(defaultValue)] [Default(defaultValue)] [...]
    [Required(reqValue)] [Flag(flagValue)]
    [RangeMin(rangeMinValue)] [RangeMax(rangeMaxValue)]
. . .
. . .
. . .

```

The `New AttributeDefList` command creates a list of attribute definitions. Each attribute definition is defined the same way as for an individual attribute definitions. You may repeat the entry for each attribute definition in the list. See “New AttributeDef” on page 93. *Structured FrameMaker only.*

Example 1:

This sample script creates a list of three attribute definitions each of which allows a string value.

```

. . .
New AttributeDefList NewVar(attrDefListVar)
  AttrDefName('MyAttrDef')
  AttrDefName('MySecondAttrDef')
  AttrDefName('MyThirdAttrDef');
. . .

```

Example 2:

This sample script creates a list of three attribute definitions of various types.

```

. . .
New AttributeDefList NewVar(attrDefListVar)
  AttrDefName('MyAttrDef1') AttrType(AtInteger)
    RangeMin(100) RangeMax(200)
  AttrDefName('MyAttrDef2') AttrType(AtString)
    Default('MyVal')
  AttrDefName('MyAttrDef3') AttrType(AtChoices)
    Choice('Choice1') Choice('Choice2') Choice('Choice3')
    Required(True);
. . .

```

New ElementLoc

Format:

```
New ElementLoc NewVar(varname)
    Parent(elementvar) Child(elementvar) [Offset(integervalue)];
```

The **New ElementLoc** command creates a element location variable. *Structured FrameMaker only.*

Table 104: New ElementLoc Options

Option Name	Option Description
Parent	This option specifies the parent element for the element location
Child	This option specifies the child element for the element location
Offset	The option specifies the offset of the element location. If not specified, zero is assumed.
NewVar	The name of the variable to hold the newly created range.

Example:

This sample script creates an element location at the first position of the structure of the currently active document.

```
. . .
SET parentEltVar = HighestLevelElement;
New ElementLoc NewVar(eltvar) Parent(parentEltVar)
    Child(parentEltVar.FirstChildElement);
. . .
```

See also

“New Structured Objects” on page 90.

New ElementRange

Format:

```
New ElementRange NewVar(varname)
    Parent(eltvar) Child(eltvar2) [Offset(offset1)]
    [Parent(eltvar3)] [Child(eltvar4)] [Offset(offset2)];
```

The **New ElementRange** command creates a element range variable. *Structured FrameMaker only.*

Table 105: New ElementRange Options

Option Name	Option Description
Parent	This option specifies the parent element for the element range. The first parent option becomes the beginning of the element range and the second parent option becomes the ending of the element range. If not specified, zero is assumed.
Child	This option specifies the child element for the element range. The first child option becomes the beginning of the element range and the second child option becomes the ending of the element range. If not specified, zero is assumed.
Offset	The option specifies the offset of the element range. The first offset is the beginning offset of the element range and the second is the ending offset of the element range. If there is no offset specified between the parent and Child options, then the first offset is zero..
NewVar	The name of the variable to hold the newly created element range.

Example:

This sample script creates an element range starting at the currently selected element for its first two children in the currently active document.

```

. . .
SET parentElt = ElementSelection.begin.child;
SET childElt = parentElt.FirstChildElement;
SET childElt2 = ChildElt.NextSiblingElement;
New ElementRange NewVar(elocvar) Parent(parentElt) Child(childElt)
    Parent(parentElt) Child(childElt2);
. . .

```

See also

“New Structured Objects” on page 90.

New Table Objects

New Table

The **New Table** command creates a new FrameMaker table.

Format:

```

New Table NewVar(varname) [DocObject(docvar) or TextLoc(locvar)]
    Format(formatname) NumCols(numberofcolumns)
    BodyRows(nbodyrows) HeaderRows(nheaderrows) FooterRows(nfooterrows);

```

Table 106: New Table Options

Option Name	Option Description
DocObject	The document object variable identifying the document to put the new table object. If not specified the currently active document will be used.
TextLoc	The text location variable identifying the place to insert the object. If not specified the insertion point of the document will be used.
Format	The format of the table to create (a table format name).
NumCols	The number of table columns to create. Default: Get from the Table format.
BodyRows	The number of table body rows to create. Default: Get from the Table format
HeaderRows	The number of table header rows to create. Default: Get from the Table format
FooterRows	The number of table footer rows to create. Default: Get from the Table format
NewVar	The name of the variable to hold the newly created table object.

Example:

This example creates a new table at the insertion point of the currently active document. The new table will have 3 columns, 4 body rows, 2 header rows and 1 footer row.

```

New Table NewVar(newTableObj)
    Format ('Format A') NumCols(3) HeaderRows(2) BodyRows(4) FooterRows(1);

```

See also

, “New TableRows” on page 98, “New TableCols” on page 98 and “Straddle TableCells Command” on page 143.

New TableRows

The **New TableRows** command adds rows to the specified table.

Format:

```
New TableRows TableObject(docvar) RowObject(rowobject)
  Direction(addirection)
  BodyRows(nbodyrows) HeaderRows(nheaderrows) FooterRows(nfooterrows);
```

Table 107: New TableRows Options

Option Name	Option Description
TableObject (<i>Required</i>)	The Table object variable identifying the table to add the new rows.
RowObject (<i>Required</i>)	The Table Row object variable identifying the starting point to add the new rows.
Direction	Which direction to add the rows. Possible values: Above Below
BodyRows	The number of table body rows to create. Default: Get from the Table format
HeaderRows	The number of table header rows to create. Default: Get from the Table format
FooterRows	The number of table footer rows to create. Default: Get from the Table format

Example:

This example adds 1 header row, 4 body rows and 1 footer row to the first defined table in the currently active document.

```
set tableobj = FirstTblInDoc;
New TableRows TableObject(tableobj)
  Direction (Below) HeaderRows(1) BodyRows(4) FooterRows(1);
```

See also

“Delete TableRows” on page 25.

New TableCols

The **New TableCols** command adds columns to the specified table.

Format:

```
New TableCols TableObject(docvar) StartCol(columnnumber)
  Direction(addirection)
  NumCols(nbodyrows);
```

Table 108: New TableCols Options

Option Name	Option Description
TableObject (<i>Required</i>)	The Table object variable identifying the table to add the new columns.
StartCol	The column number identifying the starting point to add the new columns. The first column is number 0.
Direction	Which direction to add the rows. Possible values: Left Right
NumCols	The number of table body rows to create. Default: Get from the Table format

Example:

This example adds 3 columns after the first column in the table.

```
set tableobj = FirstTblInDoc;
New TableCols TableObject(tableobj)
Direction (right) NumCols(3);
```

See also

“Delete TableColumns” on page 24.

New Text Operations

New Text

The **New Text** command inserts text into a FrameMaker text object (a paragraph or textline).

Format:

```
New Text [DocObject(docvar) or TextLoc(locvar) or Object(objvar)] [NewVar(endtloc)]
stringexpression;
```

Table 109: New Text Options

Option Name	Option Description
DocObject	The document object variable identifying the document to put the new text. If not specified and the Textloc option is not specified the insertion point of the currently active document will be used.
TextLoc	The text location variable identifying the place to insert the object. If not specified the insertion point of the document will be used.
Object	If this option is chosen, ElmScript builds a text location using this object and an offset of zero.
NewVar	Returns a TextLoc variable with the value of the end position of the new text.
stringexpression	The text to insert.

Example:

This example creates two paragraph beginning at the main flow of the currently active document and inserts some text in these new paragraphs.

```
New Paragraph PrevObject(MainFlowInDoc) NewVar(pgf1Obj);
New Paragraph PrevObject(pgf1Obj) NewVar(pgf2Obj);
New Text Object(pgf2Obj) 'Add some text right here';
New Text Object(pgf1Obj) 'Add some text to the first paragraph too';
```

See also

“Delete Text” on page 25, “Apply TextProperties” on page 8, “Get TextProperties” on page 54.

New ElmScript Data Variables

This variation of the New command is the standard way of creating a data items, data lists and data structures in ElmScript. You can create any type of allowable data using this command. Described below are methods for creating simple data values. Most of these are here for completeness only because it is easier to use the Set command (described elsewhere) for most of these. There are a few cases, however, where you need to use these commands. These deal mostly with data conversion. The Set command will evaluate the expression and determine the resulting data type from the expression. The New command allows you to determine the type of the resulting data. For example, the following two commands have expressions that evaluate to the same value.

```
Set gvIntVar = 100+300;
New Real NewVar(gvRealVar) Value(100+300);
```

The first one creates an integer value, so the value assigned to the variable will be an integer type. The second command also evaluates to an integer value, but it converts it to a real value before assigning it to the variable.

New Simple Data Types

New Integer

This commands evaluates the expression in the Value option, converts it to an integer value (if possible), then assigns it to the variable identified by the NewVar option.

Format:

```
New Integer NewVar(varname) Value(expression);
```

Table 110: New Integer Options

Option Name	Option Description
Value	The expression to evaluate. If it is not an integer data type, it will be converted, if possible.
NewVar	The name of the variable to place the result.

Example:

```
Set gvVar1 = 1.234;
Set gvVar2 = gvVar1;
New Integer NewVar(gvVar3) Value(gvVar1);
Display 'var1='+gvVar1+' var2='+gvVar2+' var3='+gvVar3;
```

This will display the following:

```
var1=1.234 var2=1.234 var3=1
```

New Real

This commands evaluates the expression in the Value option, converts it to an real value (if possible), then assigns it to the variable identified by the NewVar option.

Format:

```
New Real NewVar(varname) Value(expression);
```

Table 111: New Real Options

Option Name	Option Description
Value	The expression to evaluate. If it is not a real data type, it will be converted, if possible.
NewVar	The name of the variable to place the result.

New Metric

This commands evaluates the expression in the Value option, converts it to an metric value (if possible), then assigns it to the variable identified by the NewVar option.

Format:

```
New Metric NewVar(varname) Value(expression);
```

Table 112: New Metric Options

Option Name	Option Description
Value	The expression to evaluate. If it is not a metric data type, it will be converted, if possible.
NewVar	The name of the variable to place the result.

New String

This commands evaluates the expression in the Value or IntValue options, converts it to an string value (if possible), then assigns it to the variable identified by the NewVar option.

Format:

```
New String NewVar(varname) Value(expression) or IntValue(int expression);
```

Table 113: New String Options

Option Name	Option Description
Value	The expression to evaluate. If it is not a real data type, it will be converted, if possible.
IntValue	The integer value of the new single character string variable . This is a method for creating a single character string variable when the character is not printable.
NewVar	The name of the variable to place the result.

Example:

```
New String NewVar(gvCopyRightVar) IntValue(169);
Write Console 'ElmScript' + gvCopyRightVar;
```

This will write the following on the Frame console window:

```
ElmScript©
```

New Object

This commands evaluates the expression in the Value option (or IntValue and DocObject options), converts it to an FrameMaker Object value (if possible), then assigns it to the variable identified by the NewVar option.

Format:

```
New Object NewVar(varname) Value(expression);
or
New Object NewVar(varname) IntValue(int expression) DocObject(Fm DocObject);
```

The first format is unnecessary because there is no conversion from any other type to a FrameMaker object type. It is here for completeness only. The second format is occasionally necessary. In some cases, FrameMaker returns an integer value or an IntList of values which are really FrameObject IDs. For example, the InCond property returns a list of Condition Objects as a list of integers (IntList). In order to use these individual objects you must use this command to convert the integer into an object. Be sure to include the Document Object for a valid FrameMaker Object.

Table 114: New Object Options

Option Name	Option Description
Value	The value of the new variable. If it is not the same type as the target data, it will be converted, if possible.
IntValue	The integer value of the object value. This is a method for converting an integer into an object. Be sure that the integer is really an FrameMaker Object value before doing this or the resulting FrameMaker Object will be invalid.
DocObject	The document object (for Object types only).
NewVar	The name of the variable to hold the newly created text file object.

Example:

```

Set gvCondList = ActiveDoc.InCond;
If gvCondList.Count > 0
  New Object NewVar(gvCondObject) IntValue(gvCondList[1]) DocObject(ActiveDoc);
  Display 'First Condition Name is '+gvCondObject.Name;
Else
  Display 'The current insertion point has no conditions applied!';
EndIf

```

New Structures

New TextLoc

Format:

```

New TextLoc NewVar(gvTextLocVar)
  Object(objectvar) [Offset(integervalue)];

```

The **New TextLoc** command creates a text location variable.

Table 115: New TextLoc Options

Option Name	Option Description
Object	This option specifies the object (Pgf or Textline) for the text location
Offset	The option specifies the offset of the text location. If not specified, zero is assumed.
NewVar	The name of the variable to hold the newly created text location.

Example:

This sample script creates a text location at the first position of the first paragraph in the currently active document.

```

. . .
New TextLoc NewVar(gvTextLocVar) Object(FirstPgfInDoc);
. . .

```

New TextRange

Format:

```

New TextRange NewVar(gvTextRangeVar)
  Object(objectvar1) [Offset(offset1)] [Object(objectvar2)] [Offset(offset2)];

```

The **New TextRange** command creates a text range variable.

Table 116: New TextRange Options

Option Name	Option Description
Object	This option specifies the object (Pgf or Textline) for the text range. The first object option becomes the beginning of the text range and the second object option becomes the ending of the text range.

Table 116: New TextRange Options

Option Name	Option Description
Offset	This option specifies the offset of the text range. The first offset is the beginning offset of the text range and the second is the ending offset of the text range. If there is no offset specified between the Object options, then the first offset is zero..
NewVar	The name of the variable to hold the newly created text range.

Example:

This sample script creates a text range starting at the fourth position of the first paragraph in the currently active document and ending at the 100th position of the second paragraph.

```

. . .
New TextRange NewVar(gvTextRange) Object(FirstPgInDoc) Offset(4)
                                Object(FirstPgInDoc.NextPgInDoc) Offset(100);
. . .

```

New DataRecord**Format:**

```
New DataRecord NewVar(gvDataRecordVar) FieldName(value) [FieldName2(value2)] ... ;
```

The **New DataRecord** command creates a data record variable. The individual fields can be access by name.

Table 117: New DataRecord Options

Option Name	Option Description
FieldName	Any valid data name.
value	The value for the field.
NewVar	The name of the variable to hold the newly created value.

Example:

This sample script creates a data record with address information.

```

. . .
New DataRecord NewVar(gvDataRecord) MyName('Herman Munster')
                                Address('1313 Mockingbird Lane')
                                City('San Jose') State('California');
. . .

```

New Simple List Data

Certain data types are list types (StringList, IntList, MetricList, UIntList). These contain fixed types of data (strings, integers, metrics, Unsigned Integers, respectively). There is also a DataList type, which can contain different kinds of data. The members of these list types can be accessed by the indexing operator ([]), as follows:

```
New StringList NewVar(gvMyList) Value('String1') Value('String2') Value('String3');
Display 'The second string is '+gvMyList[2];
```

Note: These list types are used primarily for fixed length lists. ElmScript provides several other array types that are more flexible for updating. The DataList is a simple list that can have different kinds of data in each member.

It is also efficient for adding and deleting members. There are also `EslObjects` that can contain different types of data. See the `EslObject` reference for more information on the `ElmScript` array types.

The following commands are provided for creating, accessing and updating members of these list types.

New `IntList`

New `UIntList`

New `MetricList`

New `StringList`

New `DataList`

Format:

```
New {IntList UIntList MetricList StringList DataList} NewVar(varname)
    Value(expression) value(expression) ... value(expression);
```

The new list data types commands allow you to create list variables. `MetricLists` are used by `FrameMaker` for various lists of measurements, such as the widths of `Table` columns. `StringLists` are used for various name lists, including font lists and dialog scroll box lists. `UIntlists` are used to record f-code lists.

Table 118: New List Data Type Options

Option Name	Option Description
Data type	The type of data to create. Possible values are: IntList UIntList MetricList StringList DataList
Value	The value of each member of the new variable. Except for the <code>DataList</code> , if it is not the same type as the target data, it will be converted, if possible.
NewVar	The name of the variable to hold the newly created text file object.

Example 1:

This sample script sets the column widths of the first table in the document to 1 inch, 2 inches and 1.5 inches respectively.

```
New MetricList NewVar(gvColumnWidths) Value(1") Value(2") Value(1.5");
set gvTableVar = FirstTblInDoc;
set gvTableVar.TblColWidths = gvColumnWidths;
Delete Var(gvColumnWidths);
```

Example 2:

This sample script creates a list of names, then displays them in a scroll box.

```
New StringList NewVar(gvNameList) Value('George Washington')
  Value('John Adams') Value('Thomas Jefferson');
DialogBox Type(ScrollBox) Title('President Selection Dialog')
  Caption('Select a president') Init(-1) List(gvNameList)
  NewVar(gvPresidentName) Button(gvButton);
```

Example 3:

This sample script creates a list of various data types.

```
New DataList NewVar(gvList) Value(1) Value('My String') Value(ActiveDoc);
```

New PointList

Format:

```
New PointList NewVar(varname)
  X(expression) Y(expression) [X(expr) Y(expr)] ...;
```

The new PointList command allows you to create a pointlist variable. A pointlist variable is a list of points such as appears in a polygon object. Each time an X option occurs it starts a new point value. In other words the number of points in the pointlist is the same as the number of X options. The following Y option will apply to the previous X value.

Table 119: New PointList Options

Option Name	Option Description
X	The X coordinate of the point (e.g. 1").
Y	The Y coordinate of the point (e.g. 2").
NewVar	The name of the variable to hold the newly created pointlist object.

Example:

This sample script defines a pointlist, then assigns that pointlist to the Body paragraph format..

```
New Pointlist NewVar(gvPointList) X(1") Y(2") X(1.5") Y(3.0") X(2") Y(4.5");
New Line NewVar(gvLineVar) Points(gvPointList);
```

New TabList

Format:

```
New TabList NewVar(varname)
  Tab(expression) [TabType(tabtypeval)] [TabLeader(leader)] [TabDecimal(decval)]
  [Tab(expression) ...];
```

The new TabList command allows you to create a tablist variable. A tablist variable is a list of tabs such as appears in a paragraph format. Each time a Tab option occurs it starts a new tab value. In other words the number of tabs in the tablist is the same as the number of Tab options. The options that follow (TabType, TabLeader and/or TabDecimal), if any, will apply to that tab value.

Table 120: New TabList Options

Option Name	Option Description
Tab	The tab offset value (e.g. 1").
TabType	The type of the tab to create. See “Tab TabType values” on page 107:
TabLeader	The string that appears before the tab. Default NONE.
TabDecimal	The character to align the tab around. Default value ‘.’
NewVar	The name of the variable to hold the newly created tablist object.

Table 121: Tab TabType values

TabType values	TabType Description
TabLeft	Left Tab. Default value.
TabCenter	Center Tab.
TabRight	Right Tab.
TabDecimal	Decimal Tab.
TabRelativeLeft	Relative Left Tab.
TabRelativeCenter	Relative Center Tab.
TabRelativeRight	Relative Right Tab.
TabRelativeDecimal	Relative Decimal Tab.

Example:

This sample script defines a tablist, then assigns that tablist to the Body paragraph format..

```

New Tablist NewVar(gvTablist) Tab(1") TabType(TabRight) TabLeader('. ')
    TabDecimal('.',') Tab(2") Tab(2.5") Tab(3.5");
Get Object Type(PgfFmt) NewVar(gvPgfVar) Name('Body');
Set gvPgfVar.Tabs = gvTablist;

```

New PropertyList**Format:**

```

New PropertyList NewVar(varname)
    PropertyName(PropertyValue) PropertyName(PropertyValue) ... ;

```

The **New PropertyList** command creates a property list variable.

Table 122: New PropertyList Options

Option Name	Option Description
PropertyName	Specifies a valid property name. This can be a property name from any FrameMaker object.

Table 122: New PropertyList Options

Option Name	Option Description
PropertyValue	This specifies the value for the specified property name.
NewVar	The name of the variable to hold the newly created text file object.

Example

The following code creates property list variable called nprops) and adds the specified member to it:

```

. . .
New PropertyList NewVar(gvPropList) Name('CellBody') FontSize(12) KeepWithPrev(True);

Add Property To(gvPropList) PgfAlignment(PgfLeft);
. . .

```

The list will now contain four properties.

See also

“Add Property” on page 4.

New Script Variables

New LibVar

Format:

```

New LibVar
  Path(DirectoryName) NewVar(varname);

```

This creates a LibVar variable. This type of variable makes it easier to run scripts outside the current script. A directory on your hard disk (Folder on the Macintosh), can be used as a library of scripts. The file extension is added by checking the list of file extensions available.

Table 123: New LibVar Options

Option Name	Option Description
Path (Required)	The name of the directory or folder on your hard disk that will be used for the library.
NewVar (Required)	The name of the variable to hold the newly created LibVar.

Example 1:

This example creates a LibVar variable and uses it to run a script called GetInfo in the c:\ElmScript\Lib directory. The File extension (e.g. fsl) is added from the list of available file extensions (See Options in the users guide)

```

. . .
New LibVar NewVar(gvMyLib) Path('C:\ElmScript\Lib');
Run gvMyLib.GetInfo Parm1(123) Parm2('Val2') returns Val(XXX);
. . .

```

New ScriptVar

Format:

```

New ScriptVar NewVar(varname)
    File(FileName)
or
    ScriptText(StringValue)
;

```

This creates a ScriptVar variable. This type of variable makes it easier to run scripts outside the current script. You can use this variable type to run an entire file as a script or to run a subroutine inside another script.

Table 124: New ScriptVar Options

Option Name	Option Description
File	The name of the file on your hard disk that will be used for the script.
ScriptText	A string value consisting of script commands. These commands will be compiled into a script. You must use either the ScriptText option or the File option.
NewVar (Required)	The name of the variable to hold the newly created ScriptVar.

Example 1:

This example creates a ScriptVar variable and uses it to run the script `C:\ElmScript\MyScripts\GetIt.fsl`.

```

. . .
New ScriptVar NewVar(gvMyScript) File('C:\ElmScript\MyScripts\GetIt.fsl');
Run gvMyScript Parm1(123) Parm2('Val2') returns Val(XXX);
. . .

```

Example 2:

This example creates a ScriptVar variable and uses it to run a subroutine called `GetInfo` inside the `C:\ElmScript\MyScripts\Utils.fsl` file.

```

. . .
New ScriptVar NewVar(gvMyScript) File('C:\ElmScript\MyScripts\Utils.fsl');
Run gvMyScript.GetInfo Parm1(123) Parm2('Val2') returns Val(XXX);
. . .

```

Example 3:

This example builds a script from a string then runs the script. Since it runs a ScriptVar and not a subroutine, it runs the `$Main` subroutine.

```

Set gvAAA = 1;
Set gvBBB = 3;
New ScriptVar NewVar(gvMyMemoryScript)
    ScriptText('Set gvResult = gvAAA + gvBBB');
Run gvMyMemoryScript;
MsgBox 'New value is '+gvResult;

```

New SubVar

Format:

```

New SubVar
    [File(FileName)] Subname(subname) NewVar(varname);

```

This creates a SubVar variable. This type of variable makes it easier to run scripts that our not necessarily known at design time. You can use this variable type to run subroutine that may different depending on run-time considerations. It is particularly useful to pass a subvar to a another subroutine, which may, in turn, run the subvar.

Table 125: New SubVar Options

Option Name	Option Description
File	The name of the file on your hard disk that contains the script. If not specified, the current script will be used.
Subname (Required)	The name of the subroutine..
NewVar (Required)	The name of the variable to hold the newly created ScriptVar.

Example 1:

This example creates a SubVar variable and uses it to run the subroutine called Sub1.

```

. . .
New SubVar NewVar(gvMySub) SubName('Sub1');
Run gvMySub pvParm1(123) pvParm2('Val2') returns pvVal(xxx);
. . .
. . .

Sub Sub1 using pvParm1, pvParm2, pvVal
  Set pvVal = pvParm1 + 999;
  Display pvParm2;
EndSub

```

Example 2:

This example creates a SubVar variable and uses it pass the name of the subroutine to the subroutine called TestSub, which runs it.

```

. . .
New SubVar NewVar(gvMySub) SubName('Sub1');
Run TestSub pvCallBack(gvMySub) Returns pvRetVal(gvMyRetVal);
Display gvMyRetVal;
. . .
. . .

Sub Sub1 using pvParm1 pvParm2 pvVal
  Set pvVal = pvParm1 + 999;
  Display pvParm2;
EndSub

. . .
Sub TestSub using pvCallBack pvRetVal
  Local    lvBBB;
  Run pvCallBack pvParm1(444) pvParm2('qqq') returns pvVal(lvBBB);
  Set pvRetVal = lvBBB;
EndSub

```


New TextFile

New TextFile

The **New TextFile** command creates a new text data file.

Format:

```
New Textfile File(filename) NewVar(filevar) IOType(ftype) Utf8Bom(type);
```

Table 126: New Textfile Options

Option Name	Option Description
File	The file name to create.
IOType	WriteOnly Append.
Utf8Bom	If the value is 1 and the IOType is WriteOnly, this will write out a UTF-8 Byte order mark when the file is created.
NewVar	The name of the variable to hold the newly created text file object.

Example:

This example creates a text file called 'c:\temp\test.txt' and writes out two text lines before closing it.

```
New TextFile file('c:\temp\test.txt') NewVar(gvFileVar) IOType(WriteOnly);
Write Object(gvFileVar) 'Write line 1 out to File';
Write Object(gvFileVar) 'Write line 2 out to File';
Close TextFile Object(gvFileVar);
```

New EslObjects

Creating EslObjects is described in the EslObject Reference Document.

New Xml

The New Xml command creates a new Xml Document

Format:

```
New Xml File(filename) NewVar(xmlDocObj) [xmloptions];
[ReturnStatus(statusList)]
```

Table 127: New XmlOptions

Option Name	Option Description
xmlOptions	The open options are a variety of options for opening the document, controlling the handling of the many possible error conditions that might arise when documents or books are opened. These also apply when creating a new document from a template (see New the Document from a Template command). These options are described in the next table.
NewVar	The name of the variable to hold the newly opened document or book object.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the open is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned. See “Return Status for Open Document/Book” on page 116.

Table 128: Xml Options

Open Option	Value of Open Option
DocType	
Extension	
PublicId	
StructuredApplication	This option specifies the name of the Structured application to use when creating theXml file.
SystemId	
Visible	Make the document visible when it is opened. Use False to make it invisible. True or False

Open Commands

Open Project (See Es1Object reference FM 2017 or greater)

Open Document or Book

The Open document (or Book) command opens an existing FrameMaker document or book into the current FrameMaker session

Format:

```
Open Document File(filename) NewVar(docobj) [openoptions];
  [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
  [ReturnNativeError(nativeError)]
Open Book File(filename) NewVar(bookobj) [openoptions];
  [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
  [ReturnNativeError(nativeError)]
```

Table 129: Open Document(Book) Options

Option Name	Option Description
File	This option specifies the file name of the document or book.
openoptions	The open options are a variety of options for opening the document, controlling the handling of the many possible error conditions that might arise when documents or books are opened. These also apply when creating a new document from a template (see New the Document from a Template command). These options are described in the next table.
NewVar	The name of the variable to hold the newly opened document or book object.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the open is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned. See “Return Status for Open Document/Book” on page 116.
ReturnFilename	This option specifies a variable name to store a string containing the actual file name. This might be different than the specified file name if the ShowBrowser option is chosen.
ReturnNativeError	This option specifies a variable name to store an integer value that is usually the same as ErrorCode. This value is usually Zero for success, -42 for a system error, -43 for an option error or -44 for a cancelation.

The following table lists the Open Document/Book options. The default value for the option, if any, is in bold style text. For options that have True/False values, you may just use the option name to specify the True value (see import example).

IMPORTANT: Most of the time the default value for the open options will serve. You only need to specify an option and value if you want something to change.

Table 130: Open Options

Open Option	Value of Open Option
AlertUserAboutFailure	Alert the user if an unexpected condition happens. True or False
BeefyDoc	The document is a large document macintosh document. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog): (Macintosh only). OK or Cancel or DoShowDialog
BookIsInUse	The book is already open. You may cancel the command (Cancel), have the user choose (DoShowDialog), or reset the file lock and open it anyway (ResetLockAndContinue). Cancel or DoShowDialog or ResetLockAndContinue
DisallowBookDoc	Disallow the opening of FrameMaker binary books. True or False
DisallowBookMif	Disallow the opening of FrameMaker mif books. True or False
DisallowDoc	Disallow the opening of FrameMaker binary documents. True or False

Table 130: Open Options

Open Option	Value of Open Option
DisallowFilterTypes	Disallow the opening of documents via filters. True or False
DisallowMif	Disallow the opening of FrameMaker mif documents. True or False
DisallowPlainText	Disallow the opening of plain text files. True or False
DisallowSgml	Disallow the opening of SGML documents. True or False
DisallowXml	Disallow the opening of Xml documents. FM 7.0 or greater True or False
DitaMaxRefLevels	Specifies the number of reference levels to be opened while opening a DITA file. FM 2017 or greater
DoNotLockFile	Specifies whether to lock the file or not. FM 2017 or greater True or False
DontNotifyApiClient	Don't notify other Api clients about this open command. By default, FrameMaker will notify Api clients when files are opened or closed (among many other events), if that client asks to be notified. In ElmScript, you do this by including an event procedure (NotePreOpenDoc, etc). This option lets you cancel that behavior by setting the value to True. True or False
FileIsInUse	The document is already open. You may cancel the command (Cancel), have the user choose (ShowDialog), reset the file lock and open it anyway (ResetLockAndContinue), open it for view only (OpenViewOnly) or open an editable copy of the document (OpenEditableCopy). OpenViewOnly or OpenEditableCopy or DoShowDialog or ResetLockAndContinue or Cancel
FileIsOldVersion	The document was created with an older version of FrameMaker. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
FileIsStructured	The document was created with an SGML version of FrameMaker. You can remove the structure and continue (StripStructureAndOpen), cancel the operation (Cancel), open it for viewing only (OpenViewOnly), or have the user choose (DoShowDialog). OpenViewOnly or Cancel or StripStructureAndOpen or DoShowDialog
FileIsText	This file is a text only file. You may cancel the command (Cancel), open it by converting each end of line into paragraph breaks (TextFileEolIsEOP), open it without converting each end of line into a paragraph break (TextFileEolIsNotEop), or have the user choose (DoShowDialog). TextFileEolIsEOP or TextFileEolIsNotEop or DoShowDialog or Cancel
FilterFormatId	The format id of the filter to use for this open command. FrameMaker will attempt to select a filter based on the file itself. In some case, it may select incorrectly. You may specify which filter to use by including this option on the open document command. The value is from the formatId in the maker.ini file. This option is available for FrameMaker 5.5 or greater. Default: Have FrameMaker select the filter.
FontChangedMetric	FrameMaker needs to change a Font Metric in this document. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog

Table 130: Open Options

Open Option	Value of Open Option
FontNotFoundInCatalog	The catalog contains fonts that are not available on your system. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
FontNotFoundInDoc	The document contains fonts that are not available on your system. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
ForceOpenAsText	The document will be opened as a text only document. True or False
IsErrorDoc	FM 11 or greater. True or False???
LanguageNotAvailable	The document was created with another language version of FrameMaker. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
LockCantBeReset	When FrameMaker tried to open this document (or book) it could not reset the lock. The product might have crashed and the lock left open. You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog). OK or Cancel or DoShowDialog
MakeIconic	The document will initially appear as an icon instead of a window. Not available on the Macintosh. True or False
MakeVisible	Make the document visible when it is opened. Use False to make it invisible. True or False
NameStripe	String expression that will go into the title bar. Default is the file name. Not available on the Macintosh.
OpenAsType	The format of the file to import. TypeBinary - A FrameMaker binary file (standard FrameMaker file) TypeMif - A MIF file TypeText - A Text file TypeSgml - An Sgml File TypeXml - An Xml File. FM 7.0 or greater Default: Frame determines the type.
OpenBookViewOnly	The book will initially appear as a view only book. True or False
OpenDocViewOnly	The document will initially appear as a view only document (use ESC F lk to return it to the normal open document). True or False
OpenFileNotWritable	The document is not writable. You can cancel the operation (Cancel) or warn the user (DoShowDialog). Cancel or DoShowDialog
OpenId	This option lets you open this document in the same window as a currently open document. The value for this option is the document object of the window to place the opened document. The document in this window is automatically closed before the new file is opened. By default, the document goes into a new window.

Table 130: Open Options

Open Option	Value of Open Option
RefFileNotFound	The document makes references to other files (imports by reference) and FrameMaker cannot find them. You can proceed as usual (AllowAllRefFilesUnFindable), cancel the operation (Cancel) or have the user choose (DoShowDialog). AllowAllRefFilesUnFindable or Cancel or DoShowDialog
SgmlOpenApplication	This option specifies the name of the SGML application to use when opening an SGML file. This takes precedence over any other SGML application specified. FM 5.5.6 or 6.0
ShowBrowser	This option tells FrameMaker to display the open dialog box to the user. True or False
StructuredOpenApplication	This option specifies the name of the Structured application to use when opening an SGML or Xml file. This takes precedence over any other SGML or Xml application specified. FM 7.0 or greater
TemplateShouldInsertRoot	FM 11 or greater. True or False???
UpdateBrowserDirectory	Update the directory from the browser dialog box if applicable. True or False
UpdateMTOC	This option updates the mini TOC, if present. DoUserPreference or Yes or No
UpdateTextReferences	This option determines how the text insets are updated when the document is opened. You may follow the user preferences in the DontUpdateTextInsets property of the document (DoUserPreference), or automatically have the text insets updated (Yes) or don't update the text insets at all (No). DoUserPreference or Yes or No
UpdateXRefs	This option determines how the cross-references are updated when the document is opened. You may follow the user preferences in the DontUpdateXRefs property of the document (DoUserPreference), or automatically have the cross-references updated (Yes) or don't update the cross-references at all(No). DoUserPreference or Yes or No
UseAutoSaveFile	This option determines how FrameMaker will use an autosave file, if available. You may cancel the command(Cancel), proceed as usual(Yes), don't use the autosave file (No) or have the user choose (DoShowDialog). Cancel or Yes or No or DoShowDialog
UseRecoverFile	This option determines how FrameMaker will use a recover file, if available. You may cancel the command (Cancel), proceed as usual(Yes), don't use the autosave file (No) or have the user choose (DoShowDialog). Cancel or Yes or No or DoShowDialog
VerifyInsets	.FM 10 or greater

Table 131: Return Status for Open Document/Book

Status Text	Description
"AutoSaveFileUsed"	Message: Autosave file was used.
"BadFileName"	Option Error: The specified filename was invalid.
"BadFileRefsWereMapped"	Message: The file reference contained illegal characters; the illegal characters were converted to something safe for the current platform.
"BadFileType"	Option Error: The file was an executable file or other unreadable type.

Table 131: Return Status for Open Document/Book

Status Text	Description
"BadScriptValue"	Option Error: The open options contained an invalid property value.
"BadTemplate"	System Error: A bad template was specified.
"BeefyDoc"	Message: The document file was extremely large (Macintosh platforms only), but it was opened anyway.
"CancelBeefyDoc"	Canceled: The document was very large and the user or the script disallowed it (Macintosh platforms only).
"CancelBook"	Canceled: The file was a book and the Open script disallowed it.
"CancelBookMIF"	Canceled: The file was a MIF file and the Open script disallowed it.
"CancelFileBrowser"	Canceled: The user canceled the Open operation from the file browser.
"CancelFileHasStructure"	Canceled: The file had structure, so the user or the script canceled the Open operation.
"CancelFileIsDoc"	Canceled: The file was a document and the Open script disallowed it.
"CancelFileIsFilterable"	Canceled: The file was a filterable file and the Open script disallowed it.
"CancelFileIsInUse"	Canceled: The file was in use, so the user or the Open script canceled the Open operation.
"CancelFileIsMIF"	Canceled: The file was a MIF file and the Open script disallowed it.
"CancelFileIsOldVersion"	Canceled: The file was from an old version of a FrameMaker product, so the user or the Open script canceled the Open operation.
"CancelFileIsSgml"	Canceled: The file was an SGML document and the Open script disallowed it.
"CancelFileIsText"	Canceled: The file was text, so the user or the Open script canceled the Open operation.
"CancelFileIsXml"	Canceled: The file was an XML document and the Open script disallowed it.
"CancelFontMetricsChanged"	Canceled: The file contained fonts with changed metrics, so the user or the Open options canceled the Open operation.
"CancelFontsMapped"	Canceled: The document contained fonts that needed to be mapped to other fonts, so the user or the Open options canceled the Open operation.
"CancelFontsMappedInCatalog"	Canceled: The document's Character Catalog or Paragraph Catalog contained fonts that needed to be mapped to other fonts, so the user or the Open options canceled the Open operation.
"CancelFontsWithUnavailableEncodings"	Canceled: The file contained fonts with unavailable encodings, so the user or the Open options canceled the Open operation.
"CancelLanguagesNotFound"	Canceled: The file contained languages that weren't available, so the user or the Open options canceled the Open operation.
"CancelOpenFileNotWritable"	Canceled: The document is not writable.
"CancelReferencedFilesNotFound"	Canceled: The file contained referenced files that were not available, so the user or the Open options canceled the Open operation.
"CancelTempDiskFull"	System Error: There was insufficient room on the disk to cache data while opening the file.
"CancelUseAutoSaveFile"	Canceled: An Autosave file was present, so the user or the Open options canceled the Open operation.
"CancelUseRecoverFile"	Canceled: A recover file was present, so the user or the Open options canceled the Open operation.
"CantForceOpenAsText"	Option Error: Open options attempted to open the file as text, but file was wrong type.

Table 131: Return Status for Open Document/Book

Status Text	Description
"CantNewBooks"	Option Error: The options specified a book that didn't exist (the Open operation can't create a new book).
"CantOpenBooksFluid"	Canceled: The file contained a fluid view, so the user or the Open options canceled the Open operation.
"CantOpenBooksViewOnly"	Canceled: The file was a view only book, so the user or the Open options canceled the Open operation.
"ChecksumDamage"	System Error: Bad checksum on file.
"DisallowedType"	Option Error: The file was a Frame binary document and the Open options disallowed it.
"DocDamagedByTextFilter"	Option Error: The file was a text document and was damaged when it was filtered.
"DocHeadersDamaged"	System Error: The document headers were damaged (probably because of a file system problem).
"DocWrongSize"	System Error: The file is the wrong size (probably because of a file system problem).
"EditableCopyOpened"	Message: The file was in use and the user or the Open options opened an editable copy.
"FileAlreadyOpen"	Message: The file was in use and the user or the Open options opened another copy (Macintosh platforms only).
"FileAlreadyOpenThisSession"	Option Error: The file is already open and the options disallowed opening another copy.
"FileHadStructure"	Option Error: The file had FrameMaker features, but current FrameMaker product is not FrameMaker.
"FileHasNewName"	Message: The filename was changed from the name specified.
"FileIsNotWritable"	The file was not writable, and the user canceled the open via the alert.
"FileIsOldVersion"	Message: The file was from an old FrameMaker product version, but the user or the Open script chose to open it anyway.
"FileIsText"	Message: The file was a Text Only file, but the user or the Open script chose to open it anyway.
"FileIsViewOnly"	Message: The file is a View Only file.
"FileModDateChanged"	Message: The file has changed since the last time it was opened or saved in the current session.
"FileNotReadable"	System Error: You don't have read permission for the file.
"FileStructureStripped"	Message: The file had FrameMaker features, which the user or the Open script chose to strip.
"FileWasFiltered"	Message: File was filterable and was filtered.
"FileWasInUse"	Message: The file was in use.
"FontMetricsChanged"	Message: The file contained fonts with changed metrics, but it was opened anyway.
"FontsMappedInCatalog"	Message: The Paragraph or Character Catalog had unavailable fonts, which were mapped to substitute fonts.
"FontsWereMapped"	Message: The document contained unavailable fonts, which were mapped to substitute fonts.
"FontsWithUnavailableEncodingsUsed"	Message: The file contained fonts with unavailable encodings, but it was opened anyway.
"LanguagesWerentFound"	Message: The document used some unavailable languages, but it was opened anyway.

Table 131: Return Status for Open Document/Book

Status Text	Description
"LockCouldntBeReset"	Message: The file had lock that couldn't be reset.
"LockNotReset"	Message: The file had lock that wasn't reset.
"LockWasInvalid"	Message: The file had invalid lock.
"LockWasReset"	Message: The file lock was reset.
"MissingScript"	System Error: Internal Error.
"OpenedFluid"	Message: The flow in fluid view.
"OpenedViewOnly"	Message: The user or the Open options chose to open the file as a View Only file.
"RecoverFileUsed"	Message: A recover file was used.
"ReferencedFilesWerentFound"	Message: The imported graphics files couldn't be found, but the file was opened anyway.
"TooManyWindows"	System Error: Too many windows are open.
"UnresolvedTextInsets"	Message: There were unresolved text insets, but the file was opened anyway.
"UnresolvedXRefs"	Message: There were unresolved cross-references, but the file was opened anyway.
"UserCanceled"	Canceled: The user canceled the Open operation.

Example 1:

This example opens a document in the file `c:\mydocs\mydoc.fm` bypassing any messages if the template is from an older version of FrameMaker or if the fonts aren't found.

```
. . .
Open Document File ('c:\myfiles\mydoc.fm') NewVar(docobj)
    FileIsOldVersion FontNotFoundInDoc ReturnStatus(gvErrorList);
If ErrorCode not = 0
    Display 'Open Failure-'+ErrorCode+ ' Msg-'+ErrorMsg;
    Display 'Open Messages-'+gvErrorList;
EndIf
. . .
```

Example 2:

This example opens a document in the file `c:\mydocs\mydoc.fm`, putting in the same window as the currently active document. The active document is automatically closed before the new file is opened.

```
. . .
Open Document File ('c:\myfiles\mydoc.fm') NewVar(docobj) OpenId(ActiveDoc);
. . .
```

See also

“New Document” on page 76, “Close Document” on page 12, “Save Document or Book” on page 132.

Open TextFile

The **Open Textfile** command opens an existing text data file.

Format:

```
Open Textfile File(filename) NewVar(filevar) IOType(ftype) BomVar(bomType);
```

Table 132: Open Textfile Options

Option Name	Option Description
File	The file name to create.
IOType	ReadOnly WriteOnly Append.
BomVar	A variable name to which the command returns a byte order mark indicator. The value will be 0 for standard Ansi and 1 for UTF8.
NewVar	The name of the variable to hold the opened text file object.

Example:

This example opens a text file called 'c:\temp\test.txt' and writes out two text lines before closing it.

```
Open TextFile file('c:\temp\test.txt') NewVar(gvFileVar) IOType(Append);
Write Object(gvFileVar) 'Write line 1 to the end of the File';
Write Object(gvFileVar) 'Write line 2 to the end of the File';
Close TextFile Object(gvFileVar);
```

Paste Text

The **Paste Text** command pastes the current clipboard contents to the current text selection.

Format:

```
Paste Text [DocObject (docobjectvar) [pasteoptions]
```

Table 133: Paste Text Options

Option Name	Option Description
DocObject	The document object variable from which to clear the selection. If not specified, the current active is used.
Interactive	Notify user for options.
ReplaceCells	Replace table cells
VisibleOnly	Only the visible portion of the selection
DontDeleteHiddenText	Don't delete hidden text
DontApplyAllRows	Not to all rows.
InsertBelowRight	Add columns to the right.

Example

The following script pastes the clipboard contents to the current selection of the current active document with deleting hidden text:

```
. . .
Paste Text DontDeleteHiddenText;
. . .
```

See also

“Clear Text” on page 11, “Cut Text” on page 14, and “Copy Text” on page 13.

PopClipboard

The **PopClipboard** command moves the entry on top of the clipboard stack to the clipboard itself. This (in conjunction with the PushClipboard command) allows you to save and restore the original state of the clipboard, if you want to do Cut/Copy/Paste operations, but want the original clipboard contents to return when you are finished.

Format:

```
PopClipboard;
```

Example

The following script saves the current clipboard contents to the clipboard stack, copies the currently selected text to the clipboard and after the insertion point is changed, it pastes the text then restores the clipboard from the clipboard stack.

```
. . .
PushClipboard; // Save current clipboard to clipboard stack
Copy Text;    // Copy current selection to clipboard
. . .        // Do something to change insertion point
Paste Text;   // Paste the previously copied text
PopClipboard; // Restore clipboard to original state
. . .
```

See also

“PushClipboard” on page 123, “Paste Text” on page 120, and “Copy Text” on page 13

Print Commands

Print Document or Book

The **Print Document** (or Book) command prints an open document or book in the current FrameMaker session

Format:

```
Print Document DocObject(docvar);
Print Book BookObject(bookvar);
```

Table 134: Print Document(Book) Options

Option Name	Option Description
DocObject or BookObject	This option specifies the document or book or book object to print. If not specified, the currently active document is printed.

IMPORTANT: The print options for the document or book are specified by setting those properties in the document or book.

Example:

This example prints two copies of pages 45 through 60 of the document represented by the docobj variable. Blank pages will also be printed.

```

. . .
set docobj.PrintBlankPages = True;
set docobj.PrintNumCopies = 2;
set docobj.PrintStartPage = 45;
set docobj.PrintEndPage = 60;
set docobj.PrintScope = PrRange;
Print DocObject (docobj);

```

See also

“Open Document or Book” on page 112 and “Close Document” on page 12.

Promote

The **Promote** command promotes the selected structural element from the specified document. The selected element becomes a sibling of its former parent. It appears immediately after its former parent. The siblings that follow it become its children. *Structured FrameMaker only.*

Format:

```

Promote [DocObject (docvar)] [BookObject(bookvar)]
      [Element(eltvar)] [ElementRange(eltrangeVar)];

```

Table 135: Promote Options

Option Name	Option Description
DocObject	The document object variable from which to promote the element. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to promote the element. If not specified and the Doc object is also not specified, the currently active book is used.
Element	The element object to promote. If not specified, then the current element selection will be used.

Example

The following script promotes the current element selection of the currently active document.

```

. . .
Promote;
. . .

```

Example

The following script promotes the first child of the current element.

```

. . .
SET eltvar = ElementSelection.begin.child;
SET childvar = eltvar.FirstChildElement;
Promote Element(childvar);
. . .

```

See also

“Unwrap” on page 144, “Demote” on page 27, and “Wrap” on page 149.

PushClipboard

The **PushClipboard** command moves the contents of the clipboard to the clipboard stack. This (in conjunction with the **PopClipboard** command) allows you to save and restore the original state of the clipboard, if you want to do Cut/Copy/Paste operations, but want the original clipboard contents to return when you are finished.

Format:

```
PushClipboard;
```

Example

The following script saves the current clipboard contents to the clipboard stack, copies the currently selected text to the clipboard and after the insertion point is changed, it pastes the text then restores the clipboard from the clipboard stack.

```
. . .
PushClipboard; // Save current clipboard to clipboard stack
Copy Text;    // Copy current selection to clipboard
. . .        // Do something to change insertion point
Paste Text;   // Paste the previously copied text
PopClipboard; // Restore clipboard to original state
. . .
```

See also

“**PopClipboard**” on page 121, “**Paste Text**” on page 120, and “**Copy Text**” on page 13

Quit Session

The **Quit Session** command terminates the current FrameMaker session, which also terminates the program. Any ‘Terminate’ or ‘close doc’ events are run before the product actually stops running, giving you a chance to cancel this operation.

Format:

```
Quit Session;
```

Read command

The **Read** command reads a line of text from a text file.

Format

```
Read File(filevar) NewVar(stringvar) [Rewind] [MaxLine(LineSize)];
```

Table 136: Read Options

Option Name	Option Description
File	The file variable of the file to read.
NewVar	A string variable where the text line will be placed.
Rewind	Specifies that the text file will be positioned at the beginning.
LineSize	Specifies the maximum size of the text line. This is not necessary unless you are reading especially long text lines. The standard buffer size is 512 characters. You do not have to use this option unless your text lines are greater than 511 characters. Most text lines are under 100 characters.

Use the `ErrorCode` variable to determine the results of the read command. If the value is zero then it worked. If it is positive, then it reached the end of the file. A negative value is (as usual) an error.

Example:

This example opens a text file called 'C:\ElmScript\test.txt', reads each line of the file and writes the text to the console. Then, it rewinds the file to the beginning and does it again.

```

Set gvInFileName = 'C:\TestData\test.txt';
Set ErrorCode = 0;
Open Textfile File(gvInFileName) NewVar(gvTextFile) IOType(ReadOnly);
If ErrorCode not = 0
    Display 'Open error on text file ' + ErrorMessage;
    LeaveSub;
EndIf

Read File(gvTextFile) NewVar(gvTextBuffer);
Loop While (ErrorCode = 0)
    Write console gvTextBuffer;
    Read File(gvTextFile) NewVar(gvTextBuffer);
EndLoop

Read File(gvTextFile) Rewind;
Read File(gvTextFile) NewVar(gvTextBuffer);
Loop While (ErrorCode = 0)
    Write console gvTextBuffer;
    Read File(gvTextFile) NewVar(gvTextBuffer);
EndLoop

Close TextFile Object (gvTextFile);

Display 'Done';

```

Remove Commands

Remove Attribute

The **Remove Attribute** command removes an attribute that has no value from the specified structural elements.

Format

```
Remove Attribute(AttrName) [DocObject(docvar)] [BookObject(bookvar)]
  From(elementvar);
```

Table 137: Remove Attribute Options

Option Name	Option Description
DocObject	The document object variable from which to remove the undefined attribute. If not specified, the currently active document is used.
BookObject	The book object variable from which to remove the undefined attribute. If not specified, the currently active book is used.
Attribute (Required)	The name of the attribute to remove.
From (Required)	Specifies the object from which to remove the attribute. If the attribute is required or has a default value, it won't be removed..

Example 1

The following script removes the 'Type' attribute the parent element in the current element selection.

```
. . .
Set eltVarParent = ElementSelection.begin.Parent;
Remove Attribute('Type') From(eltVarParent);
. . .
```

See also

“Remove UndefAttr” on page 128.

Remove CommandObject

The **Remove CommandObject** command removes an existing user menu command (or menu separator) or a FrameMaker menu command (or menu separator) from an existing menu object.

Format:

```
Remove CommandObject(commandobjectvarname) From(menuobjectvarname)
```

Table 138: Remove CommandObject Options

Option Name	Option Description
CommandObject (Required)	The variable name of the command object (or menu separator object)
From (Required)	The variable name of the menu object.

The `commandobjectvarname` is created by the **New Command** command. The `menuobjectvarname` is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu. NOTE: This command removes the command object from the menu but it does NOT delete the command itself. Also, it only removes the command from the specified menu. To make it unavailable to the user, you must remove it from all the menus

Example:

The following script removes a command with the name 'UserCmd' from the File menu.

```

. . .
Get Object Type(Menu) Name('FileMenu') NewVar(filemenuvar);
Get Object Type(Command) Name('UserCmd') NewVar(usercmdvar);
Remove CommandObject(usercmdvar) From(filemenuvar);
. . .

```

See also

“New Menu Commands” on page 84, “Add CommandObject” on page 1, and “Get Object” on page 46

Remove Member

The **Remove Member** command removes an individual member from a list data item.

Format

```
Remove Member[ (membervalue) ] [Number(membernumber)] From(listvar);
```

Table 139: Remove Member Options

Option Name	Option Description
Member	Value of the member to remove.
Number	The integer number of the member in the list. The first member is 1.
From (Required)	Specifies the list variable name.

Example 1

The following code removes member number 2 in the string list:

```

. . .
New StringList NewVar(gvFruitNameList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Remove Member Number(2) From(gvFruitNameList);
. . .

```

The list will now contain two members, 'Apples' and 'Peaches'.

Example 2

The following code removes the member 'Peaches' from the string list:

```

. . .
New StringList NewVar(gvFruitNameList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Remove Member('Peaches') From(gvFruitNameList);
. . .

```

The list will now contain two members, 'Apples', and 'Oranges'.

Remove MenuObject

The **Remove MenuObject** command removes an existing user menu or a FrameMaker menu from an existing menu or menu bar.

Format:

```
Remove MenuObject(menuobjectvarname1) Frmo(menuobjectvarname2)
```

Table 140: Remove MenuObject Options

Option Name	Option Description
MenuObject (<i>Required</i>)	The variable name of the menu object.
From (<i>Required</i>)	The variable name of the menu or menu bar object

The menuobjectvarname1 is created by the **New Menu** command. The menuobjectvarname2 is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu.

Example:

The following script removes a menu named 'MyMenu' from the main FrameMaker menu bar.

```
. . .
Get Object Type(Menu) Name('!MakerMainMenu') NewVar(makermainvar);
Get Object Type(Menu) Name('MyMenu') NewVar(mymenuvar);
Remove MenuObject(mymenuvar) From(makermainvar);
. . .
```

See also

“New Menu” on page 85, “Add MenuObject” on page 3, and “Get Object” on page 46.

Remove MenuSepObject

The **Remove MenuSepObject** command removes an existing menu separator or a FrameMaker separator from an existing menu.

Format:

```
Remove MenuSepObject(sepobjectvarname) From(menuobjectvarname)
```

Table 141: Remove MenuSepObject Options

Option Name	Option Description
MenuSepObject (<i>Required</i>)	The variable name of the separator object.
From (<i>Required</i>)	The variable name of the menu object.

The sepobjectvarname is created by the **New MenuSeparator** command or retrieved by the **Get Object** command. The menuobjectvarname is created by the **New Menu** command or retrieved by the **Get Object** command for an existing menu.

Example:

The following script removes a menu separator named 'MySepItem' from the FrameMaker Table menu in the main maker menu bar.

```

. . .
Get Object Type(Menu) Name('TableMenu') NewVar(maketablevar);
Get Object Type(MenuItemSeparator) Name('MySepItem') NewVar(mysepvar);
Remove MenuSepObject(mysepvar) From(maketablevar);
. . .

```

See also

“New MenuItemSeparator” on page 86, “Add MenuSepObject” on page 3, and “Get Object” on page 46.

Remove Property

The **Remove Property** command removes an individual property from a property list data item.

Format

```
Remove Property(PropertyName) From(proplistvar);
```

Table 142: Remove Property Options

Option Name	Option Description
Property	The name of the property to remove
From (Required)	Specifies the property list variable name.

Example 1

The following code creates a property list then removes the fontsize member from it:

```

. . .
new propertylist NewVar(nprops) Name('CellBody') FontSize(12) KeepWithPrev(True);
Remove Property(fontsize) From(nprops);
. . .

```

The list will now contain two members, CellBody and KeepWithPrev.

See also

“New PropertyList” on page 107, and “Add Property” on page 4.

Remove UndefAttr

The **Remove UndefAttr** command removes an attribute that has no value from the specified structural elements.

Format

```
Remove UndefAttr(AttrName) [DocObject(docvar)] [BookObject(bookvar)]
From(objvar);
```

Table 143: Remove UndefAttr Options

Option Name	Option Description
DocObject	The document object variable from which to remove the undefined attribute. If not specified, the currently active document is used.
BookObject	The book object variable from which to remove the undefined attribute. If not specified, the currently active book is used.
UndefAttr (Required)	The name of the undefined attribute to remove.
From (Required)	Specifies the object from which to remove the undefined attribute. If this object is an element object, the undefined attribute is removed from that element. If the object is an element definition, the undefined attribute is removed from all elements of that type. If the object is a document or book, then the undefined attribute will be removed from all elements in the document or book.

Example 1

The following script removes the 'Type' attribute if it is undefined from all 'Chapter' elements.

```

. . .
Get Object Type(ElementDef) NewVar(gvEltDefVar) Name('Chapter');
Remove UndefAttr('Type') From(gvEltDefVar);
. . .

```

See also

“Demote” on page 27, and “Unwrap” on page 144.

Replace Member

The **Replace Member** command replaces an individual member from a list data item.

Format

```
Replace Member[(membervalue)] [Number(membernumber)] In(listvar) With(newmembervalue);
```

Note: This command is now obsolete. It is kept for backward compatibility with previous versions. You can replace members easier and more efficiently using the index operator (`[]`).

Table 144: Replace Member Options

Option Name	Option Description
Member	Value of the member to replace.
Number	The integer number of the member in the list. The first member is 1.
In (Required)	Specifies the list variable name.
With	Value of replacement member.

Example 1:

The following code replaces member number 2 in the string list:

```
. . .
New StringList NewVar(gvFruitNameList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Replace Member Number(2) In(gvFruitNameList) With('Grapes');
. . .
```

The list will now contain three members, 'Apples', 'Grapes', and 'Peaches'.

Example 2:

The following code replaces member number 2 in the string list. This is the same as the above except that it uses the indexing operator instead of the Replace Member command

```
. . .
New StringList NewVar(gvFruitNameList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Set gvFruitNameList[2] = 'Grapes';
. . .
```

Example 3:

The following code replaces member 'Peaches' in the string list with 'Grapes':

```
. . .
New StringList NewVar(gvFruitNameList)
    Value('Apples') Value('Oranges') Value('Peaches');
. . .
Replace Member('Peaches') In(gvFruitNameList) With('Grapes');
. . .
```

The list will now contain three members, 'Apples', 'Oranges', and 'Grapes'.

Return Command

The **Return** command allows you to send a message back from a notification event. This is only applicable for event scripts, while processing a notification event.

Format:

```
Return {Cancel Value(integerexpression) SkipStep};
```

Table 145: Return Options

Option Name	Option Description
Cancel	This option tells FrameMaker to cancel the current notification operation. For example, you may cancel a document open command, by using this option during a NotePreOpenDoc event. Note: You must do this during the NotePRExxxxx notification event. The NotePOSTxxxxx event occurs after the command has been completed!
Value	This option tells ElmScript to send an integer back to the client which caused this notification. This applies only for the NoteCallClient notification event.
SkipStep	If you execute the return command using this option during a NotePreSaveAsPDF notification event, the Acrobat dialog is not displayed. <i>This option is available for FrameMaker version 5.5 and greater.</i>

Table 145: Return Options

Option Name	Option Description
DisplayedXRefDialog	This return value tells FrameMaker that you are using your own XRef dialog. FrameMaker will send a notification (NoteDisplayClientXRefDialog) to a client when <i>This option is available for FrameMaker version 9.0 and greater.</i>
ClosedXRefDialog	If you execute the return command using this option during a NotePreSaveAsPDF notification event, the Acrobat dialog is not displayed. <i>This option is available for FrameMaker version 9.0 and greater.</i>

Example:

This script sample cancels the open document request, if the user tries to open a document called 'c:\temp\tempfile.fm'.

```

. . .
Event NotePreOpenDoc
  if Filename = 'c:\temp\tempfile.fm'
    Return cancel;
  EndIf
EndEvent
. . .

```

Run Command

The **Run** command allows you to run ElmScript subroutines. A subroutine is a defined set of command (see “Sub Command” on page 143). The subroutine runs the subroutine commands until it gets to the end of the subroutine or a LeaveSub command is executed. You may pass data names to a subroutine, which the subroutine may use during its run. You may also pass data names to the subroutine, which the subroutine may modify. **NOTE:** Though you may pass properties names and constant values *to* a subroutine, you may only return variable names *from* a subroutine.

Format:

```
Run subExpression [Parm1[(value1)] ... ParmN[(valueN)] [returns retparm(varname)];
```

Table 146: Run Options

Option Name	Option Description
subExpression	This is an expression that evaluates to a subroutine (SubVar data type). Many times this is just the name of the subroutine (from the Sub command). If the expression is a string value, the value in the string value will be used as the subroutine name.
Parm	These options allow you to pass information to the subroutine. This can be any name that is not otherwise in use. You don't have to use the word Parm. If the value is not present, then the name of the parm is passed to the subroutine as an integer variable with the value of True (1).
returns	This option specifies that the following parm name is modifiable in the subroutine. It must be a variable name and not a value.

Save Commands

Save Document or Book

The **Save Document** (or Book) command allows you to save a document or book back to disk after making changes. This command also provides the Save As function if supply the file option. You may save the document into any valid file name you choose. You may save a document in a variety of forms, including normal FrameMaker binary, Mif, PDF, and many others. The FrameMaker 5.5 (or greater) version of ElmScript allows you to choose a filter id and save a document into any of the provided output filters. .

Format:

```
Save Document [DocObject(docvar)] [File(newfilename)] [saveoptions];
  [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
  [ReturnNativeError(nativeError)]
Save Book [BookObject(bookvar)] [File(newfilename)] [saveoptions];
  [ReturnStatus(statusList)] [ReturnFilename(returnFilename)]
  [ReturnNativeError(nativeError)]
```

Table 147: Save Document or Book Options

Option Name	Option Description
DocObject	The document object variable identifying the document to save. If not specified the currently active document will be used.
BookObject	The book object variable identifying the book to save.
File	The new file name. This is used for the Save As function. To save the document or book with the same file name, omit this option.
saveoptions	The save options are a variety of options for saving the document, controlling the handling of the many possible error conditions that might arise when documents or books are saved. These options are described in the next table.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the save is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned. See “Return Status Values for Save Document/Book” on page 138.
ReturnFilename	This option specifies a variable name to store a string containing the full path name.
ReturnNativeError	This option specifies a variable name to store an integer value that is usually the same as ErrorCode. This value is usually Zero for success, -42 for a system error, -43 for an option error or -44 for a cancelation.

Table 148: Save Options

Save Option	Value of Save Option
AlertUserAboutFailure	Alert the user if an unexpected condition happens. True or False
AutoBackupOnSave	This option determines how the backup files are treated when the document is saved. You may follow the user preferences in the AutoBackup property of the document (SaveUserPrefAutoBackup), or automatically make the backup file (SaveYesAutoBackup) or don't create a backup file at all (SaveNoAutoBackup). SaveUserPrefAutoBackup or SaveYesAutoBackup or SaveNoAutoBackup
BleedBottom	Metric value. FM 2019 or greater
BleedInside	Metric value. FM 2019 or greater
BleedMarks	True or False . FM 2019 or greater
BleedOutside	Metric value. FM 2019 or greater
BleedTop	Metric value. FM 2019 or greater
ColorBars	True or False . FM 2019 or greater
CropMarks	True or False . FM 2019 or greater
DitaApplyOutputTemplates	DoUserPreference or Yes or No . FM 2019 or greater
DitaBookChapterTemplate	FM 2019 or greater
DitaBookChapterTitleTemplate	FM 2019 or greater
DitaBookIndexTemplate	FM 2019 or greater
DitaBookLOFTemplate	FM 2019 or greater
DitaBookLOTTemplate	FM 2019 or greater
DitaBookTOCTemplate	FM 2019 or greater
DitaBookSectionTemplate	FM 2019 or greater
DitaBookSectionTitleTemplate	FM 2019 or greater
DitaBookSubsectionTemplate	FM 2019 or greater
DitaBookSubsectionTitleTemplate	FM 2019 or greater
DitaBookTitleTemplate	FM 2019 or greater
DitaCompositeDocTemplate	Specifies which template to use while saving the DITA map as PDF or a book with FrameMaker components. FM 2017 or greater
DitaGenerateComponentsAtOneLoc	DoUserPreference or Yes or No . FM 11 or greater
DitaGenerateFlatBook	DoUserPreference or Yes or No . FM 10 or greater
DitaGenerateBookTitlePage	DoUserPreference or Yes or No . FM 2019 or greater
DitaGenerateFolderTitlePage	DoUserPreference or Yes or No . FM 2019 or greater
DitaGenerateIndex	DoUserPreference or Yes or No . FM 2019 or greater
DitaGenerateLOF	DoUserPreference or Yes or No . FM 2019 or greater

Table 148: Save Options

Save Option	Value of Save Option
DitaGenerateLOT	DoUserPreference or Yes or No. <i>FM 2019 or greater</i>
DitaGenerateTOC	DoUserPreference or Yes or No. <i>FM 2019 or greater</i>
DitaOptimizePostProcess	DoUserPreference or Yes or No. <i>FM 2019 or greater</i>
DitaPostProcessingOnBook	DoUserPreference or Yes or No. <i>FM 11 or greater</i>
DitaSavePdfViaBook	The Ditaval Condition tag. <i>FM 11 or greater</i>
DitavalCondTag	The Ditaval Condition tag. <i>FM 10 or greater</i>
DitavalFile	The Ditaval File. <i>FM 9.0 or greater</i>
DontNotifyApiClients	<p>Don't notify other Api clients about this save command. By default, FrameMaker will notify Api clients when files are saved, if that client asks to be notified. In ElmScript, you do this by including an event procedure (NotePreSaveDoc, etc). This option lets you cancel that behavior by setting the value to True.</p> <p>True or False</p>
FileIsInUse	<p>Someone else is using the document. You may cancel the command(Cancel), have the user choose (DoShowDialog), or reset the file lock and open it anyway (ResetLockAndContinue)</p> <p>OpenViewOnly or OpenEditableCopy or DoShowDialog or ResetLockAndContinue or Cancel</p>

Table 148: Save Options

Save Option	Value of Save Option
FileType	<p>The type of file to save:</p> <p>SaveFmtBinary - Normal FrameMaker binary format for the current version.</p> <p>SaveFmtInterchange - Mif format</p> <p>SaveFmtInterchange70 - Save in Mif 7.0 format (FM 8.0 or greater)</p> <p>SaveFmtInterchange80 - Save in Mif 8.0 format (FM 8.0 or greater)</p> <p>SaveFmtInterchange90 - Save in Mif 9.0 format (FM 9.0 or greater)</p> <p>SaveFmtInterchange100 - Save in Mif 10.0 format (FM 10 or greater)</p> <p>SaveFmtInterchange110 - Save in Mif 11.0 format (FM 11 or greater)</p> <p>SaveFmtInterchange120 - Save in Mif 12.0 format (FM 12 or greater)</p> <p>SaveFmtInterchange130 - Save in Mif 13.0 format (FM 2015 or greater)</p> <p>SaveFmtInterchange140 - Save in Mif 14.0 format (FM 2017 or greater)</p> <p>SaveFmtInterchange150 - Save in Mif 15.0 format (FM 2019 or greater)</p> <p style="text-align: center;">IMPORTANT: Not all of the above values are valid for all versions of Framemaker. Some are capable of only the most recent version and one before.</p> <p>SaveFmtViewOnly - View Only binary format</p> <p>SaveFmtText - Plain text format</p> <p>SaveFmtPdf - PDF format</p> <p>SaveFmtSgml - Sgml Format</p> <p>SaveFmtXml - Xml Format (FM 7.0 or greater)</p> <p>SaveFmtBinary60 - Save in FM 6.0 format (FM 7.0 or greater)</p> <p>SaveFmtBinary70 - Save in FM 7.0 format (FM 8.0 or greater)</p> <p>SaveFmtBinary80 - Save in FM 8.0 format (FM 8.0 or greater)</p> <p>SaveFmtBinary90 - Save in FM 9.0 format (FM 9.0 or greater)</p> <p>SaveFmtBinary100 - Save in FM 10.0 format (FM 10 or greater)</p> <p>SaveFmtBinary110 - Save in FM 11.0 format (FM 11 or greater)</p> <p>SaveFmtBinary120 - Save in FM 12.0 format (FM 12 or greater)</p> <p>SaveFmtBinary130 - Save in FM 13.0 format (FM 2015 or greater)</p> <p>SaveFmtBinary140 - Save in FM 14.0 format (FM 2017 or greater)</p> <p>SaveFmtBinary150 - Save in FM 15.0 format (FM 2019 or greater)</p> <p style="text-align: center;">IMPORTANT: Not all of the above values are valid for all versions of Framemaker. Some are capable of only the most recent version and one before.</p> <p>SaveFmtFilter - Use a filter to save the document. You must supply a filter id for this option.</p> <p>SaveFmtCompositeDoc - Save composite doc. FM 9.0 or greater</p> <p>SaveFmtBookWithXml - Save Book with Xml. FM 9.0 or greater</p> <p>SaveFmtBookWithFm - Save Book with FM. FM 9.0 or greater</p>

Table 148: Save Options

Save Option	Value of Save Option
FilterFormatId	<p>The format id of the filter to use for this open command. FrameMaker will attempt to select a filter based on the file itself. In some case, it may select incorrectly. You may specify which filter to use by including this option on the open document command. The value is from the <code>formatId</code> in the <code>maker.ini</code> file.</p> <p>This option is available only for <i>FrameMaker 5.5 and greater</i>.</p> <p>Use 'HTML' to save as html and use 'XML' to save as xml (<i>FrameMaker 5.5.6 or greater</i>).</p> <p>Default: Have FrameMaker select the filter.</p>
LockCantBeReset	<p>When FrameMaker tried to save this document (or book) it could not reset the lock. The product might have crashed and the lock left open. It might have a permissions problem (under Unix). You can proceed as usual (OK), cancel the operation (Cancel) or have the user choose (DoShowDialog).</p> <p>OK or Cancel or DoShowDialog</p>
MakePageCount	<p>How to round the page count, during the save operation. You may have FrameMaker use the document <code>PageRounding</code> property (<code>UseCurrentSetting</code>), leave the page count alone (<code>DontChangePageCount</code>), make the page count an even number (<code>MakePageCountEven</code>), make the page count an odd number (<code>MakePageCountOdd</code>), or remove the extra pages at the end of the document (<code>DeleteEmptyPages</code>).</p> <p><code>UseCurrentSetting</code> or <code>DontChangePageCount</code> or <code>MakePageCountEven</code> or <code>MakePageCountOdd</code> or <code>DeleteEmptyPages</code></p>
MarksOffset	Metric value. <i>FM 2019 or greater</i>
MarksWeight	Metric value. <i>FM 2019 or greater</i>
PageHeight	Metric value. <i>FM 2019 or greater</i>
PageInfo	Integer value. <i>FM 2019 or greater</i>
PageWidth	Metric value. <i>FM 2019 or greater</i>
PDFColorSpace	Integer value. <i>FM 2019 or greater</i>
PDFCompatibility	StringList value. <i>FM 2019 or greater</i>
PDFEmbedPageThumbnails	Integer value. <i>FM 2019 or greater</i>
PDFLayout	Integer value. <i>FM 2019 or greater</i>
PDFOpenPage	Integer value. <i>FM 2019 or greater</i>
PDFOptimizedWebView	Integer value. <i>FM 2019 or greater</i>
PDFPageRangeEnd	String value. <i>FM 2019 or greater</i>
PDFPageRangeStart	String value. <i>FM 2019 or greater</i>
PDFPages	Integer value. <i>FM 2019 or greater</i>
PDFPassword	String value. <i>FM 2019 or greater</i>
PDFPreset	String value. <i>FM 2019 or greater</i>
PDFPrimaryOutput	Integer value. <i>FM 2019 or greater</i>
PDFStandard	String value. <i>FM 2019 or greater</i>
PDFTag	Integer value. <i>FM 2019 or greater</i>
PDFUseDistiller	Integer value. <i>FM 2019 or greater</i>
PDFView	Integer value. <i>FM 2019 or greater</i>

Table 148: Save Options

Save Option	Value of Save Option
PDFZoom	StringList value. FM 2019 or greater
ModDateChanged	Someone else has modified this file since you opened it. You may cancel the operation (Cancel), proceed anyway (OK) or allow the user to choose (DoShowDialog). This is available for <i>FrameMaker 5.5 or greater</i> . OK or Cancel or DoShowDialog
NoStructuredErrorLog	Integer value FM 9.0 or greater The document makes references to other files (imports by reference) and FrameMaker cannot find them. You can proceed as usual (AllowAllRefFilesUnFindable), cancel the operation (Cancel) or have the user choose (DoShowDialog). AllowAllRefFilesUnFindable or Cancel or DoShowDialog
PublishLogFilePath	FM 2019 or greater
RefFileNotFound	The document makes references to other files (imports by reference) and FrameMaker cannot find them. You can proceed as usual (AllowAllRefFilesUnFindable), cancel the operation (Cancel) or have the user choose (DoShowDialog). FM 10 or greater AllowAllRefFilesUnFindable or Cancel or DoShowDialog
RegMarks	Integer value. FM 2019 or greater
RetainNameStripe	Keep the original name or change it to the one you saved it to. Use True for keeping the old name and False for using the new name. True False
SaveFileNotWritable	The document is not writable. Usually this is a permissions problem. You can cancel the operation (Cancel) or warn the user (DoShowDialog). Cancel or DoShowDialog
SaveTextExtraBlankLineAtEOP	Tells FrameMaker whether to add an extra line at the end of each paragraph for Text Only saves. Use True to add an extra line, use False otherwise. True False
SaveTextTblSetting	Tells FrameMaker how to save Text Only tables: Use the SaveTblUserPref value to use the last setting. Use the SaveTblRowsAsPgfs value to save each cell as a separate paragraph. Use the SaveTblColsAsPgfs value to save cells as paragraphs by columns, and the SaveSkipTbIs value to not output tables at all. SaveTblUserPref or SaveTblRowsAsPgfs or SaveTblColsAsPgfs or SaveSkipTbIs
SgmlSaveApplication	This option specifies the name of the SGML application to use when saving a structured document as an SGML file. This takes precedence over any other SGML application specified. (FM 5.5.6 or FM 6.0)
ShowSaveTextDialog	Tells FrameMaker whether to display the dialog box on text only saves. Use True to show it, False to not show it. True False

Table 148: Save Options

Save Option	Value of Save Option
ShowBookErrorLogPublishing	<i>FM 2019 or greater.</i> True False
ShowProgressBar	<i>FM 2019 or greater.</i> True False
SlugArea	Integer value. <i>FM 2019 or greater</i>
StructuredSaveApplication	This option specifies the name of the Structured application to use when saving a structured document as an SGML or Xml file. This takes precedence over any other SGML or Xml application specified. FM 7.0 or greater
UNIXpermissions	Integer expression, default 0666. This applicable to Unix systems only.
UpdateFRVList	Tells FrameMaker whether to add this file to the files recently visited list. Use True to add it, False to not add it. True False
UseDefaultUNIXpermissions	Tells FrameMaker whether to use the default UNIX permissions. Use True to use them, and False to use the permissions specified in the UNIXPermissions option. This applicable to Unix systems only. True False

Table 149: Return Status Values for Save Document/Book

Status Text	Description
"BadFileId"	The file's operating system ID was bad.
"BadSaveFileName"	The specified file name is not allowed by the operating system.
"BadSaveObjectId"	Bad Document or Book Object.
"BadSaveScriptValue"	The options specified has an invalid value.
"CancelSaveExportedFileInvalid"	<i>FM 10 or greater.</i>
"CancelSaveFileIsInUse"	The file is in use and the user did not or could not reset the lock. Or the file is in use, and the FileIsInUse option is set to DoCancel, or it is set to ResetLockAndContinue but the FrameMaker product could not reset the lock.
"CancelSaveFileNotWritable"	Save canceled because it was not writable.
"CancelSaveModDateChanged"	The file has changed since the last time it was opened or saved in the current session.
"FileIsNotWritable"	File is not writeable.
"FileIsViewOnly"	File is view only and cannot be saved.
"FileModDateChanged"	File mod date has changed since the last time it was opened.
"FileNotWritable"	The file was not writable.

Table 149: Return Status Values for Save Document/Book

Status Text	Description
"FileWasExported"	
"FileWasInUse"	The file was in use.
"InvalidSaveFilter"	
"LockCouldntBeReset"	The file lock couldn't be reset.
"LockNotReset"	The file lock was not reset.
"LockWasInvalid"	
"LockWasReset"	The file lock was reset.
"NonPortableFileRefs"	
"NonPortableSaveName"	The file had a name that isn't portable (Macintosh only).
"ProductIsMaker"	
"ProductIsViewer"	
"Unstructured"	
"UserCanceledSave"	The file was not saved due to user cancelation.

Example:

This example saves the document represented by the `docobj` variable, first under its original name, then under Microsoft word (using the filter id, 0490) using the name `test.doc`, then as html using the name `test.html`, then as Mif using the name `test.mif`, as plain text using the name `test.txt`, as a view only document using the name `testview.fm` and finally as a PDF document using the name `test.pdf`. The Word, HTML, and PDF save options are available in *FrameMaker 5.5 or greater*

```

. . .
Save Document DocObject(docVar) ReturnStatus(gvErrorList);
If ErrorCode not = 0
    Display 'Save Failure-'+ErrorCode+' Msg-'+ErrorMsg;
    Display 'Save Messages-'+gvErrorList;
EndIf

Save Document DocObject(docVar) File('test.doc')
    FileType(SaveFmtFilter) FilterFormatId('0490')

Save Document DocObject(docVar) File('test.html')
    FileType(SaveFmtFilter) FilterFormatId('HTML')

Save Document DocObject(docVar) File('test.fm')
    FileType(SaveFmtBinary)

Save Document DocObject(docVar) File('test.mif')
    FileType(SaveFmtInterchange)

Save Document DocObject(docVar) File('test.txt')
    FileType(SaveFmtText)

Save Document DocObject(docVar) File('testview.fm')
    FileType(SaveFmtViewOnly)

```

```
Save Document DocObject(docVar) File('test.pdf')
  FileType(SaveFmtPdf)
```

See also

“Open Document or Book” on page 112 and “Close Document” on page 12.

Select Commands

Select TableCells

The Select TableCells command selects a range of cells in a specified table.

Format:

```
Select TableCells TableObject(tablevar)
  [TopRow(startrow)] [BottomRow(endrow)] [LeftCol(startcol)] [RightCol(endcol)]
  [SelectTable];
```

Table 150: Select TableCells Options

Option Name	Option Description
TableObject (<i>Required</i>)	The Table object variable identifying the table to select the cells.
TopRow	The row number of the top most cell to select. The first row is number is 0.
BottomRow	The row number of the bottom most cell to select. The first row is number is 0.
LeftCol	The column number of the left most cell to select. The first column is number is 0.
RightCol	The column number of the right most cell to select. The first column is number is 0.
SelectTable	This is an option for selecting the entire table.

Example:

This sample script selects a range of cells from row 0 to row 3, and column 0 to column 4 of the first table in the document.

```
set tblobj = FirstTblInDoc;
Select TableCells TableObject(tblobj) BottomRow(3) RightCol(4);
```

See also

“New Table” on page 97 and “Straddle TableCells Command” on page 143.

Set command

The **Set** command assigns the value of an expression to a variable name or property. The data type of the variable will be the same as the expression. If the assignment is to a property, the expression is converted to the type of data required by the property. If the assignment is to a variable name and the variable does not already exist, this command will create a new variable in the global data space, unless the **DeclareVarMode** session variable is set to True. In this case, the command will fail.

Format:

```
Set varname = expression;
set property = expression;
set varname.property = expression;
```

Table 151: Set Options

Option Name	Option Description
varname (<i>Required</i>)	The name of an existing variable or a new variable name. ElmScript will create a new variable if it doesn't already exist, unless the DeclareVarMode session variable is set to True.
property	When a property name appears as the target of a Set command, the property must be either a session property, a document property (of the currently active document) or a book property (of the currently active book).
varname.property	The property of an object.
expression	The expression to compute. This can be any valid arithmetic or string expression, using variables and/or properties.

Example:

This example sets the session's active document to the document represented by the variable docvar. Sets the page numbering style of the document represented by docvar and then performs an absolutely meaningless computation.

```
. . .
Set ActiveDoc = gvDocVar;
Set gvDocVar.pageNumStyle = PageNumAlphaLC;
set var1 = (10 * gvDocVar.FirstPageNum + 100) / 5;
. . .
```

The **Set** command and the control commands are the exception to the above format. The **Set** command assigns a new value to a property or data variable. It will also create a data variable if it does not already exist. The control commands (**If**, **Loop**, and **Run**) work with blocks of commands which terminate with a termination command (**EndIf**, **EndLoop**, **EndSub**). See FrameMaker Reference for more information.

Sort command

The **Sort** command allows you to sort the items in a StringList or IntList variable.

Format:

```
Sort List(listVar) [NewVar(sortedList)]
    [Ascending | Descending] [Case] [Indirect] [CmpType(type)]
```

Table 152: Sort Options

Option Name	Option Description
List (<i>Required</i>)	The name of an existing StringList or IntList variable. This identifies the list to be sorted.
Ascending	Sort the list in normal character set order. This is the default value.
Descending	Sort the list in reverse character set order.
Case	Do a case sensitive sort (for StringList type only). The default is case insensitive.
Indirect	This tells the sort routine to do an indirect sort. This means that instead of producing a sorted list, it produces an IntList which contains the relative numbers of the items in sorted order. This is useful, if you don't want to change the original order in the list.
CmpType	Specifies the type of string comparison to use. A value of 0 (zero) is the default value and it will use the standard string comparison. A value of 1 indicates an alternate string comparison method. This is only useful using FrameMaker 8 in UTF-8 mode and sorting string values. The alternate comparison method is much faster but it may not perform a correct comparison for some UTF-8 characters.
NewVar	The name of the variable to put the result of the sort. If not specified, the list is sorted and the result placed back into the same variable.

Example:

The following script creates and sorts the values in a string list.

```

. . .
New StringList NewVar(gvList) value ('CCC') value('AAA') Value('BBB');
Sort List(gvList);
. . .

```

Split

The **Split** command splits structural element containing the insertion point into two elements. *Structured FrameMaker only.*

Format:

```
split [DocObject (docvar)] [BookObject(bookvar)];
```

Table 153: Split Options

Option Name	Option Description
DocObject	The document object variable from which to split the element. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to split the element. If not specified and the Doc object is also not specified, the currently active book is used.

Example

The following script splits element at the current insertion point of the currently active document.

```

. . .
split;
. . .

```

See also

“Wrap” on page 149 and “Unwrap” on page 144.

Straddle TableCells Command

The **Straddle TableCells** command allows you to cause a range of table cells to be straddled or unstraddled.

Format:

```

Straddle TableCells CellObject(cellobjvar)
    [NumRows(rowcount)] [NumCols(colcount)]
    {On Off};

```

Table 154: Straddle TableCells Options

Option Name	Option Description
CellObject (<i>Required</i>)	The Table cell object variable identifying the cell to start the straddle.
NumRows	The number of rows to straddle or unstraddle.
NumCols	The number of columns to straddle or unstraddle.
Type	Whether to Straddle or unstraddle. The value <i>On</i> means to Straddle and the value <i>Off</i> means to unstraddle.

Example:

The example finds the first cell in the first table in the currently active document and straddles two rows and two columns starting at that cell.

```

. . .
Set gvFirstCellVar = FirstTblInDoc.FirstRowInTbl.FirstCellInRow;
Straddle TableCells CellObject(gvFirstCellVar) NumRows(2) NumCols(2) On
. . .

```

See also

“New Table” on page 97 and “Select TableCells” on page 140.

Sub Command

The following shows the format of the Sub command.

Format:

```

Sub subname [using arg1[ arg2]...[ argN]];
. . .
EndSub

```

In between the Sub and EndSub lines there can be any number of commands.

Table 155: Sub Options

Option Name	Option Description
subname (<i>Required</i>)	The name of the subroutine. This name has to be unique within the script. You cannot have two subroutines with the same name in the same script file.
using	An optional filler word to enhance readability.
argI	The name of an argument. These are names (identifiers) of your choosing. Each name represents the name of a variable in the parameter data space for this subroutine. These are used as variable names inside the subroutine. These names are for documentation purposes and are treated as comments by ElmScript. The Run command supplies the actual names that are inserted into the parameter data space.
EndSub (<i>Required</i>)	The command that terminates a subroutine.

Uninstall Script

The **Uninstall script** command deregisters a script from the ElmScript system. If the script is an event script the 'terminate' event is run first then the script is removed from memory. If it is a standard script, the menu item (under the ElmScript -> Scripts menu) is removed. Note: The user may use a menu item to uninstall a script.

Format:

```
Uninstall Script Name(scriptname);
```

Table 156: Uninstall Script Options

Option Name	Option Description
Name (<i>Required</i>)	The internal name of the script. Every script has a name. If it wasn't given during the install command, ElmScript made up a name for it. If you plan to uninstall a script you need to give it a name so you can use it here to uninstall it.

Example:

The example uninstalls the event script called TableTest, which was previously installed.

```
Uninstall Script Name('TableTest');
```

Unwrap

The **Unwrap** command removes the selected structural element or elements from the specified document, but leaves their contents and child elements intact. *Structured FrameMaker only.*

Format:

```
Unwrap [DocObject (docvar)] [BookObject(bookvar)]
      [Element(eltvar)] [ElementRange(eltrangeVar)];
```

Table 157: Unwrap Options

Option Name	Option Description
DocObject	The document object variable from which to unwrap the element. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to unwrap the element. If not specified and the Doc object is also not specified, the currently active book is used.
Element	The element object to unwrap. If not specified and the ElementRange is also not specified, then the current element selection will be used.
ElementRange	The range of elements to unwrap. If not specified and the Element is also not specified, then the current element selection will be used.

Example

The following script unwraps the current element selection of the currently active document.

```
. . .
Unwrap;
. . .
```

Example

The following script unwraps the first child of the currently selected element.

```
. . .
Set gvEltVar = ElementSelection.begin.child;
Set gvChildElt = gvEltVar.FirstChildElement;
Unwrap Element(gvChildElt);
. . .
```

See also

“Wrap” on page 149, “Remove UndefAttr” on page 128 and “Demote” on page 27.

Update commands

Update objTypes

The **Update** objTypes command allows you to update a variety of document wide information.

Format:

```
Update {ReDisplay Formatting Hyphenating ResetEquationSettings ResetRefFrames
RestartPgfnNums TextInset(textinsetvar) Variables XRefs}
[DocObject(docvar)]
[ForceUpdate Internal OpenDocs ClosedDocs Exeverything];
```

Table 158: Update Options

Option Name	Option Description
Update Type (Required)	The update type indicates which update function to perform. See the next table for a list and description of the various types of updates allowed.
DocObject	This option identifies the document object where the update will take place. If not specified the currently active document will be used.
XRef options	When updating cross references, you may use any combination of the following sub-options. See the update cross-reference update options table below. <code>ForceUpdate Internal OpenDocs ClosedDocs Everything</code>

Table 159: Update Types

Update Type	Update Type Description
ReDisplay	Redisplay the document. If you turn off the <code>Displaying</code> property of a document, you should run this command to display any changes made.
Formatting	Reformat the document. You should run this command if you turn off the <code>Reformatting</code> session property and turn it back on again.
Hyphenating	Rehyphenate the document. When the user changes hyphenization information, you should run this command, to rehyphenate the document.
ResetEquationSettings	Resets the document equations to default settings.
ResetRefFrames	Resets the reference frames in the document. This is useful if you have changed a reference frame. Run this function to update the document.
RestartPgfNums	Restart the paragraph numbering for a document.
TextInset	Updates the specified text inset. You must supply the text inset object variable for this command. You can also specify a document object and it will update all the automatic text insets in the document.
Variables	Updates all the variables in the document. This is the same as clicking the Update button in the Variables dialog box.
XRefs	Updates the cross-references in the document. You may include additional options for this function. See the next table for more information.

Table 160: Cross-reference update options

XRef Update Option	XRef Update Option Description
ForceUpdate	Updates all cross-references, whether they have changed or not.
Internal	Update internal cross-references
OpenDocs	Update cross-references in open documents
ClosedDocs	Update cross-references in closed documents.
Everything	Update Internal and external cross-references

Example:

The following set of commands updates various things in the currently active document.

```

. . .
Update Display;
Update Formatting;
Set gvTiVar = FirstTiInDoc;
Update TextInset(gvTiVar);
Update XRefs OpenDocs ClosedDocs;
. . .

```

Update Book

The update Book command updates an existing FrameMaker book. *Frame 6.0 or higher*

Format:

```

Update Book BookObject(gvBookVar) [updateOptions]
      [ReturnStatus(statusList)];

```

Table 161: Update Book Options

Option Name	Option Description
BookObject	This option specifies book to update.
updateOptions	The open options are a variety of options for opening the document, controlling the handling of the many possible error conditions that might arise when documents or books are opened. These also apply when creating a new document from a template (see New the Document from a Template command). These options are described in the next table.
ReturnStatus	This option specifies a variable name to store a stringlist value containing a list of status messages returned after the Update Book is complete. These could be errors or just warnings or it could be an empty list. If this is not specified, then the information is not returned.

The following table lists the Update options. The default value for the option, if any, is in bold style text. For options that have True/False values, you may just use the option name to specify the True value.

IMPORTANT: Most of the time the default value for the update options will serve. You only need to specify an option and value if you want something to change.

Table 162: Update Book Options

Open Option	Value of Open Option
AlertUserAboutFailure	Alert the user if an unexpected condition happens. True or False
AllowInconsistentNumProps	Allow Frame to update numbering, text insets, etc. even if there are documents in the book with numbering properties that don't match the properties specified in the book. You can proceed as usual (DoOK), cancel the operation (DoCancel) or have the user choose (DoShowDialog). DoOK or DoCancel or DoShowDialog

Table 162: Update Book Options

Open Option	Value of Open Option
AllowNonFMFiles	Allow Frame to update numbering, text insets, etc. even if there are documents in the book that were not created in Frame. You can proceed as usual (DoOK), cancel the operation (DoCancel) or have the user choose (DoShowDialog). DoOK or DoCancel or DoShowDialog
AllowViewOnlyFiles	Allow Frame to the book even if there are view only documents in the book. You can proceed as usual (DoOK), cancel the operation (DoCancel) or have the user choose (DoShowDialog). DoOK or DoCancel or DoShowDialog
MakeVisible	Make the document visible when it is opened. Use False to make it invisible. True or False
OpenOldVersionFiles	Open files for a previous framemaker version. <i>FM 2015 or greater</i> True or False
ShowBookErrorLog	Display the book error log for this update. True or False
UpdateBookGeneratedFiles	Update generated files, such as TOC, indexes, etc. True or False
UpdateBookInlineComponents	Update mini table of contents in all documents within the book. <i>FM 2017 or greater</i> True or False
UpdateBookMasterPages	Update automatic assignment of master pages in all the book's documents. <i>Frame 7.0 or greater</i> True or False
UpdateBookNumbering	Update numbering in all documents. True or False
UpdateBookOLELinks	Update OLE links in all documents. True or False
UpdateBookTextReferences	Update text insets in all the book's documents. True or False
UpdateBookXRefs	Update cross-references in all the book's documents. True or False

Table 163: Return Status Values for Update Book

Status Text	Description
"BadUpdateBookFileId"	The specified book Object is invalid.
"BadUpdateBookScriptValue"	The update book options contained an invalid property value.
"BookNotSelfConsistent"	The book is not self-consistent (book generates data in one file that is source data for another generated file, or page count continually changes for this operation).
"CancelInconsistentNumPropsInFileInBook"	One or more of the book's document files has numbering properties that are inconsistent with the properties stored in the book.
"CancelNonFMFileInBook"	One or more of the book's document files is not a FrameMaker product file.

Table 163: Return Status Values for Update Book

Status Text	Description
"CancelViewOnlyFileInBook"	One or more of the book's document files is view-only.
"DuplicateFileInBook"	One or more files in the book is a duplicate of another file.
"FileInBookNotOpened"	One or more of the book's document files could not be opened.
"FileInBookNotSaved"	One or more of the book's document files could not be saved.
"NoNonGeneratedFilesInBook"	The only files in the book are generated files.
"TooManyWindowsUpdateBook"	Too many windows were open for the currently available memory.
"UserCanceledUpdateBook"	User canceled the update operation.

Example:

This example update the book and show the error log.

```

. . .
Update Book BookObject(gvBookVar) ShowBookErrorLog(True)
  ReturnStatus(gvErrorList);
If ErrorCode not = 0
  Display 'Update Book Failure-'+ErrorCode+ ' Msg-'+ErrorMsg;
  Display 'Update Book Messages-'+gvErrorList;
EndIf
. . .

```

See also

“New Document” on page 76, “Close Document” on page 12, “Save Document or Book” on page 132.

Wrap

The **Wrap** command inserts a structural element around the selected text and structural element or elements from the specified document. *Structured FrameMaker only.*

Format:

```

Wrap [DocObject (docvar)] [BookObject(bookvar)]
  {ElementDef(eltdefvar) or ElementDefName(stringvar)}
  [Element(eltvar)] [ElementRange(eltrangeVar) or TextRange(textRangeVar)];

```

Table 164: Wrap Options

Option Name	Option Description
DocObject	The document object variable from which to wrap the element. If not specified and the Book object is also not specified, the currently active document is used.
BookObject	The book object variable from which to wrap the element. If not specified and the Doc object is also not specified, the currently active book is used.
ElementDef	The element definition object with which to wrap the selected element or elements.
ElementDefName	The name of the element definition object with which to wrap the selected element or elements. You must specify the ElementDef or the ElementDefName.

Table 164: Wrap Options

Option Name	Option Description
Element	The element object to wrap. If not specified and the ElementRange is also not specified, then the current element selection will be used.
ElementRange	The range of elements to wrap. If not specified and the Element or TextRange is also not specified, then the current element selection will be used.
TextRange	The range of text to wrap. If not specified and the Element or Element range is also not specified, then the current element selection will be used.

Example

The following script wraps the current element selection of the currently active document using the 'Head' element definition.

```

. . .
Get Object NewVar(gvEltDefVar) Type(ElementDef) Name('Head');
Wrap ElementDef(gvEltDefVar);
. . .

```

Example

The following script wraps the first child of the currently selected element using the 'DBody' element.

```

. . .
Set gvEltVar = ElementSelection.Begin.Child;
Set gvChildEltVar = gvEltVar.FirstChildElement;
Wrap Element(gvChildEltVar) ElementDefName('DBody');
. . .

```

See also

“Unwrap” on page 144, “Remove UndefAttr” on page 128 and “Split” on page 142.

Write Command

The Write command allows you to write a string expression to one of three output targets: the FrameMaker console, the FrameMaker display (dialog box), or a textfile.

Format:

```
Write {Console Object(filevar)} Stringexpression;
```

Table 165: Write Options

Option Name	Option Description
Destination (Required)	The destination of the output string. This can be console to write the FrameMaker console or object(filevar) to write the string expression to a text file..
stringexpression	Any expression which can be converted into a string..

Example:

This example write two messages to the console, then creates and writes two lines to a text file called `ofile.txt`.

```
Write Console 'I am going to write a string to the console';
Write Console 'Then I will write to a text file';
New TextFile NewVar(gvTextFileVar) File('ofile.txt');
Write Object(gvTextFileVar) 'Write a line to a file';
Write Object(gvTextFileVar) 'Write another line to this file';
Close Textfile Object(gvTextFileVar);
```


Chapter 2

FrameMaker Object Reference

This chapter lists FrameMaker/ElmScript objects and their properties. Read-only properties are indicated. All other properties are updatable.

If there is a specific range of valid values for a property, it is indicated in the Description column. If a property must specify an Object, the object type is indicated. Any other special information about the property is located in the description column.

Attribute Conditional Expression

Object Name - AttrCondExpr

FM 8.0 or greater

Table 166: AttrCondExpr Properties

Property Name	Data Type	Description
AttrCondExprIsActive	Integer	True if active. False otherwise.
AttrCondExprStr	String	TheAttribute Conditional Expression.
NextAttrCondExprInDoc	Object	Next Attribute Conditional Expression Object

Books

FrameMaker books have two related objects, the Book and the BookComponent. The Book object represents the entire book file and properties associated with the entire book. The BookComponent Object represents a document within a book and its properties. The actual document properties are part of the Document object. The BookComponent properties deal mostly with book generation and print properties (when printing an entire book).

Object name - Book

Table 167: Book Properties (Page 1 of 6)

Property Name	Data Type	Description
AcrobatBookmarkDisplayTags	Integer	True if Include Paragraph Tags in Bookmark Text is on (the paragraph tag is added before the paragraph text in each bookmark).
BookDontUpdateReferences	Integer	False if FrameMaker updates cross-references when it opens the book.
BookIsModified (Read-Only)	Integer	True if the book has been modified.
BookIsSelected	Integer	True if the book icon in the book window has been selected. <i>FM 6.0 or greater</i>
BookIsViewOnly	Integer	True if the book is view only. <i>FM 6.0 or greater</i>
CustomElementList	StringList	List of tags to display when ElementCatalogDisplay is set to ElcatCustom
DocAcrobatColumnArticleThreads	Integer	True if you want separate article threads for each column, False if you want separate article threads for each text frame. Note that DocAcrobatNoArticleThreads must be false.
DocAcrobatDefaultsChanged	Integer	True if the default heuristics for determining the paragraph level are disabled.
DocAcrobatElementList	StringList	List of the element tags and context labels to include in bookmarks. This applies only to structured FrameMaker documents.
DocAcrobatElements	Integer	True if elements rather than paragraphs are used for bookmarks. This applies only to structured FrameMaker documents.
DocAcrobatNoArticleThreads	Integer	True if you do not want article threads in the resulting PDF.
ElementCatalog	EltCat	List of elements in the catalog
ElementCatalogDisplay	Integer	Catalog display options: ElcatStrict - valid children for working start to finish ElcatLoose - valid children for working in any order ElcatChildren - children allowed anywhere in parent ElcatAll - all elements ElcatCustom - the list of tags specified by the CustomElementList property
ElementSelection	ElementRange	The currently selected element range in the book.
FileInfoPacket	String	The XMP Metadata. See "XMP Metadata" on page 194. (FM 7.0 or greater)
FirstComponentInBook	Object	First component in the book (BookComponent).
FirstElementDefInDoc	Object	First element definition Object in the book (ElementDef Object).
FirstFmtChangeListInDoc	Object	First format change list Object in the book (FmtChangeList Object).
FirstSelectedComponentInBook	Object	First selected component in the book (BookComponent).

Table 167: Book Properties (Page 2 of 6)

Property Name	Data Type	Description
GenerateAcrobatInfo	Integer	True if Generate Adobe Acrobat Data is on. To generate PDF data, you must set other book print properties as follows PrintToFile: True PrintThumbnails: False PrintSeps: False PrintBlankPages: True PrintLastSheetFirst: False PrintNumCopies: 1 PrintOddPages: True PrintEvenPages: True PrintScale: 100%
HighestLevelElement	Object	Highest level element Object in the book (Element Object).
IsIconified	Integer	True if the book window is iconified (UNIX and Windows only).
IsInFront	Integer	True if the book window is in front of other windows in the FrameMaker session.
IsOnScreen	Integer	True if the book is visible on the screen. Note that this property is always True for books, and setting it to False has no effect.
Label	String	The title in the book window title bar.
Name	String	Pathname of the book.
NewElemAttrDisplay	Integer	Specifies the attribute display properties for new elements. AttrDispNone: Don't display attributes AttrDispReqSpec: display required and specified attributes AttrDispAll: display all attributes
NewElemAttrEditing	Integer	Specifies when the edit attribute dialog box appears. AttrEditNone: AttrEditRequired: AttrEditAlways:
NextOpenBookInSession (Read-Only)	Object	Next open book in session's list of open books (Book Object).
PDFAllNamedDestinations	Integer	True if PDFs generated from this book will have named destinations for every paragraph and SGML element. If False, then the generated PDF document will have named destinations only for those paragraphs and structural elements that have already been marked with PDFDestsmarked property set to True. <i>FM 6.0 or greater</i>
PDFBookmarksOpenLevel	Integer	The level of bookmarks to have expanded when Acrobat opens the generated PDF document. This can be any integer or one of the following values: PDFBookmarksOpenDefaultLevel PDFBookmarksOpenAllLevels PDFBookmarksOpenNoneLevel FM 7.0 or greater
PDFConvertCMYKtoRGB	Integer	When True this corresponds with setting Convert CMYK colors to RGB in the Save As PDF dialog box. Only has effect on Macintosh. FM 7.0 or greater

Table 167: Book Properties (Page 3 of 6)

Property Name	Data Type	Description
PDFDestsMarked	Integer	True if PDFs generated from this book will have named destinations for every paragraph and SGML element. <i>FM 6.0 or greater</i>
PDFDistillerAbsent	Integer	Set to 1 if there is no Acrobat Distiller available. FM 7.0 or greater
PDFDocInfo	StringList	A list of strings containing the values to be set for the PDF dictionary. <i>FM 6.0 or greater</i>
PDFEndPage	String	The last page of the printing page range, in the FrameMaker numbering style. .FM 7.0 or greater
PDFJobOption	String	The name of the distiller Job options. If the specified name does not exist in the Distiller Job Options list then the first Distiller Job Option in the list is used. .FM 7.0 or greater
PDFJobOptionsAbsent	Integer	Set to 1 if there is no PDF Job Options available. .FM 7.0 or greater
PDFOpenPage	String	The PDF page number, in the FrameMaker numbering style, at which Acrobat opens the generated PDF document. .FM 7.0 or greater
PDFPageHeight	Metric	The Page height for the generated PDF document. .FM 7.0 or greater
PDFPageWidth	Metric	The Page width for the generated PDF document. .FM 7.0 or greater
PDFPrintPageRange	Integer	True for generating PDF for the specified page range. False otherwise. .FM 7.0 or greater
PDFRegistrationMarks	Integer	Registration marks for the generated PDF document. Use one of the following values: PDFRegistrationMarksNone PDFRegistrationMarksWestern PDFRegistrationMarksTombo .FM 7.0 or greater
PDFSeparateFiles	Integer	True for generating a separate PDF document for each document in the book. False otherwise. .FM 7.0 or greater
PDFStartPage	String	The first page of the printing page range, in the FrameMaker numbering style. .FM 7.0 or greater
PDFZoomFactor	Metric	When PDFZoomType is set to PDFZoomNone, this is the zoom percentage of the PDF document. .FM 7.0 or greater

Table 167: Book Properties (Page 4 of 6)

Property Name	Data Type	Description
PDFZoomType	Integer	The PDF zoom setting with which the Acrobat opens the generate PDF document. Can be one of the following values: PDFZoomDefault PDFZoomPage PDFZoomWidth PDFZoomHeight PDFZoomNone .FM 7.0 or greater
PrintBlankPages	Integer	True if PageRounding allows empty pages at the end of documents.
PrintCollated	Integer	True if Collate is enabled.
PrintDownloadAsianFonts	Integer	Undocumented option.
PrintDownloadTrueTypeAsType1	Integer	Undocumented option.
PrintEmulsion	Integer	Direction of print emulsion: EmulUp: Emulsion side up EmulDown: Emulsion side down
PrinterName	String	Name of printer. Setting PrinterName on Windows has no effect. When you set PrinterName, you can set the printer to the default printer by specifying ZERO.
PrintEvenPages	Integer	True if Print Even-Numbered Pages is enabled.
PrintFileName	String	Filename of file to print to. When you set PrintFileName, you can set the filename to the default filename by specifying ZERO. Setting this property on the Macintosh has no effect.
PrintImaging	Integer	Type of print imaging: ImgPositive ImgNegative
PrintLastSheetFirst	Integer	True if Last Sheet First is enabled.
PrintLowRes	Integer	True if Low-Resolution is enabled.
PrintNumCopies	Integer	Number of copies to print.
PrintOddPages	Integer	True if Odd-Numbered Pages is enabled.
PrintPaperHeight	Metric	Height of paper.
PrintPaperWidth	Metric	Width of paper.
PrintRegistrationMarks	Integer	True if Registration Marks is enabled.
PrintScale	Integer	Scale factor expressed as a percentage of black (metric 0% to 100%)
PrintSeps	Integer	True if Print Separations is enabled.
PrintToFile	Integer	True if Print Only to File is enabled. Setting this property on the Macintosh has no effect.
PrintTomboMarks	BoolT	True if registration marks are enabled, and set to Tombo. When printing Tombo Marks, you must also set PrintRegistrationMarks to True.
ScreenHeight	Integer	Height of the book window in pixels.

Table 167: Book Properties (Page 5 of 6)

Property Name	Data Type	Description
ScreenWidth	Integer	Width of the book window in pixels.
ScreenX	Integer	The offset of the book window in pixels from the left side of the screen (or the top of the FrameMaker application window on Windows). If you set a value that would result in the book window being off the screen, that value is ignored and the old value is retained.
ScreenY	Integer	The offset of the book window in pixels from the top of the screen (or the top of the FrameMaker application window on Windows). If you set a value that would result in the book window being off the screen, that value is ignored and the old value is retained.
SeparateInclusions	Integer	True if inclusions are listed separately in the element catalog.
ServerState	Integer	Undocumented option
ServerUrl	String	Undocumented option
SgmlApplication	String	String the identifies the name of the SGML application associated with the book. (FM 5.5.6 or FM 6.0)
ShowElementDescriptiveNames	Integer	True/False. FM 10 or greater
SkipBlankSeps	Integer	True if Skip Blank Separations is enabled (don't print blank color separations).
StatusLine	String	String that appears in the book status bar. Note that this property always returns an empty string; it is effectively write-only. If you set <code>StatusLine</code> to a string other than an empty string (' '), the string will remain in the status bar until you reset it. To reset <code>StatusLine</code> so FrameMaker automatically updates the status line with normal status information, set it to an empty string (' ').
StructuredApplication	String	String the identifies the name of the SGML or Xml application associated with the book. FM 7.0 or greater
TypeOfDisplayText	Integer	Type of text for each icon in the book window. BkFileName displays the book component file name. BkText displays the first paragraph of the first flow
UseInitialStructure	Integer	True if Frame inserts initial structure for new elements.
UseInitialStructureOfAutoInsertedElements	Integer	True/False. FM 10 or greater
ViewOnlyWinBorders	Integer	True if the book has normal window borders
ViewOnlyWinMenubar	Integer	True if the book has menu bar
ViewOnlyWinPopup	Integer	True if the book has a popup menu available
XmlDoc	Integer	True indicates that the current document corresponds to an XML document. FM 7.0 or greater
XmlDocType	String	The DOCTYPE parameter from the source XML. (FM 7.0 or greater)
XmlEncoding	String	The encoding parameter of the XML Declaration for the source XML. The string is empty if no encoding is specified. If this property is set, the XML Declaration will contain the encoding parameter with this value on Save As XML. (FM 7.0 or greater)
XmlFileEncoding	String	The encoding that was detected for the source XML book. If no encoding was specified for the source XML, <code>XmlEncoding</code> will be an empty string. In that case, if this string is set, it will determine the encoding to use when saving as XML. If <code>XmlEncoding</code> has a value, this string may be empty. (FM 7.0 or greater)

Table 167: Book Properties (Page 6 of 6)

Property Name	Data Type	Description
XmlPublicId	String	The DOCTYPE public identifier for the source XML. (FM 7.0 or greater)
XmlStandAlone	Integer	An integer that specifies the XML standalone parameter for the XML document that was the source of this document. It can be one of the following values. XmlStandAloneYes XmlStandAloneNo XmlStandAloneNA This is declared in the XML Declaration. For a file with no XML declaration, the value is XmlStandAloneNoDec. For an XML declaration with no standalone parameter, the value is XmlStandAloneNone. (FM 7.0 or greater)
XmlStyleSheet	String	The XML stylesheet processing instruction to write out to XML when saving the book as XML. This value is not verified for the correct syntax. The string you set should not include the PI delimiters <? and ?>. For example, the string you supply for my.css would be: <pre>'type="text\css" href="my.css"'</pre> Only use this string to set a specific stylesheet specification. This value always returns an empty string. To get a list of stylesheet specifications associated with a book, use XmlStyleSheetList, below. (FM 7.0 or greater)
XmlStyleSheetList	StringList	A list of stylesheet processing instruction for the current book. One book can have more than one stylesheet specification associated with it. These values are not verified for the correct syntax. The strings should not include the PI delimiters <? and ?>. For example, the string you supply for my.css would be: <pre>'type="text\css" href="my.css"'</pre> Setting this property completely overwrites the current list. (FM 7.0 or greater)
XmlSystemId	String	The DOCTYPE system identifier for the source XML. (FM 7.0 or greater)
XmlUseBOM	Integer	An integer that specifies whether a byte order mark was detected when opening the source XML. It can be one of the following values. XmlUseBOMYes XmlUseBOMNo XmlUseBOMUtf8 XmlUseBOMUtf16BE XmlUseBOMUtf16LE XmlUseBOMUtf32BE XmlUseBOMUtf32LE When saving as XML, this is set to XmlUseBOMYes. FrameMaker writes a byte order mark in the resulting XML. (FM 7.0 or greater)
XmlVersion	String	The XML version that was specified in the XML Declaration when the file was opened. If no XML version was specified, this will be an empty string. (FM 7.0 or greater)
XmlWellFormed	Integer	An integer that specifies whether the source XML qualified as well formed. It can be one of the following values. XmlWellFormedYes XmlWellFormedNo XmlWellFormedNA (FM 7.0 or greater)

Object name - BookComponent

The book component represents the information in the individual components of a book file.

Table 168: Book Component Properties (Page 1 of 5)

Property Name	Data Type	Description
BookComponentType	Integer	Type of book component: BkIndexAuthor: index of authors BkIndexFormats: index of formats BkIndexMarker: index of markers BkIndexReferences: index of references BkIndexStan: standard index BkIndexSubject: subject index BkListFigure: list of figures BkListFormats: list of formats BkListMarker: list of markers BkListMarkerAlpha: alphabetical list of markers BkListPgf: list of paragraphs BkListPgfAlpha: alphabetical list of paragraphs BkListReferences: list of references BkListTable: list of tables BkNotGeneratable: book component is not a generated file BkToc: table of contents
BookComponentIsGeneratable (Read-Only)	Integer	True if book component is a generated file (BookComponentType is not set to BkNotGeneratable)
BookComponentParent (Read-Only)	Object	The parent component in the book component (BookComponent). FM 9.0 or greater
BookComponentSequenceLevel (Read-Only)	Integer	FM 11 or greater
BookComponentTemplatePath	String	The template path of the book component. FM 9.0 or greater
BookComponentTitle	String	The title of the book component. FM 9.0 or greater
BookParent (Read-Only)	Object	Book that contains the component (Book Object)
ChapNumComputeMethod	Integer	Chapter numbering type: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous chapter. NumRestart : Use the value specified for ChapterNumber . NumSame : Use the same chapter number as for the previous file. NumReadFromFile : Use the numbering properties from the document associated with this book component.
ChapterNumber	Integer	If the ChapNumComputeMethod property is NumRestart , then use this value as the chapter number. <i>FM 6.0 or greater</i>

Table 168: Book Component Properties (Page 2 of 5)

Property Name	Data Type	Description
ChapterNumStyle	Integer	The numbering style: <i>FM 6.0 or greater</i> NumStyleNumeric : Arabic. NumStyleRomanUC : Roman, uppercase. NumStyleRomanLC : Roman, lowercase. NumStyleAlphaUC : Alphabetic, uppercase. NumStyleAlphaLC : Alphabetic, lowercase. NumStyleKanji : Kanji. NumStyleZenkaku : Zenkaku. NumStyleZenkakuUC : Zenkaku, uppercase. NumStyleZenkakuLC : Zenkaku, lowercase. NumStyleKanjiKazu : Kazu. NumStyleDaiji : Daiji. NumStyleText : Text
ChapterNumText	String	If the ChapterNumStyle is equal to NumStyleText , then this property is the chapter number string. <i>FM 6.0 or greater</i>
ComponentDisplayText	String	When the associated book property TypeOfDisplayText has the value of BkText , then this is the text that displays in the book window. <i>FM 6.0 or greater</i>
ComponentElement	Object	Component element (Element Object) of this book component.
ComponentIsSelected	Integer	True if the component is selected in the book window
ComponentType (Read-Only)	Integer	The type of the component. This should be one of the following values: FM 9.0 or greater BkFile, BkFolder, BkGroup.
CorrespondingXmlPath	String	. FM 10 or greater
DocSequenceLevel (Read-Only)	Integer	FM 11 or greater
ExcludeBookComponent	Integer	The True of the component is excluded, False otherwise. FM 9.0 or greater
ExtractElementTags	StringList	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)
ExtractTags	StringList	List of paragraph tags that are used to set up a generatable file (for example, table of contents, list of figures, or list of tables)
FirstComponentInBookComponent	Object	First component in the book component (BookComponent). FM 9.0 or greater
FirstPageNum	Integer	When the PageNumComputeMethod property has the value NumRestart , then this value is used for the number of the first page. <i>FM 6.0 or greater</i>
FnCustNumString	String	Custom document footnote number text. <i>FM 6.0 or greater</i>
FnFirstNum	Integer	When the FnNumComputeMethod property has the value of NumRestart, then this value is used for the first footnote. <i>FM 6.0 or greater</i>

Table 168: Book Component Properties (Page 3 of 5)

Property Name	Data Type	Description
FnNumComputeMethod	Integer	Footnote Computing Method: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous file. NumRestart : Use the number specified by FnFirstNum. NumPerPage : Restart numbering on each page. NumReadFromFile : Use the numbering properties from the document associated with this book component.
FnNumStyle	Integer	Footnote numbering style: <i>FM 6.0 or greater</i> FnNumNumeric : Arabic FnNumRomanUC : Roman uppercase FnNumRomanLC : Roman lowercase FnNumAlphaUC : Alphabetic uppercase FnNumAlphaLC : Alphabetic lowercase FnNumKanji : Kanji characters FnNumZenkaku : Zenkaku FnNumZenkakuUC : Zenkaku uppercase FnNumZenkakuLC : Zenkaku lowercase FnNumKanjiKazu : Kazu FnNumDaiji : Daiji FnNumCustom : Custom numbering
GenerateInclude	Integer	True if the document appears in the scroll list of files to be generated by the Generate/Update command for the book
ImportFmtInclude	Integer	True if the book component is included in the list of components to be updated with imported formats when the user or a client executes Import Formats.
InsertLinks	Integer	True if hypertext links are automatically inserted in generated files
Name	String	Pathname of document that the component represents
NextBookComponentInDFSOrder	Object	Next component in the book file (BookComponent), traversing through the book components in a hierarchal manner (top to bottom). FM 9.0 or greater
NextComponentInBook	Object	Next component in the book file (BookComponent)
NextSelectedComponentInBook (Read-Only)	Object	Next selected component in the book window (BookComponent)
PageNumbering	Integer	The component document's page numbering type: Obsolete for FM 6.0 BkContPageNum BkResetPageNum BkReadFromFile
PageNumComputeMethod	Integer	Page Computing Method: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous file. NumRestart : Use the number specified by FirstPageNum. NumReadFromFile : Use the numbering properties from the document associated with this book component.

Table 168: Book Component Properties (Page 4 of 5)

Property Name	Data Type	Description
PageNumStyle	Integer	Page numbering style: <i>FM 6.0 or greater</i> PageNumNumeric : Arabic PageNumRomanUC : Roman uppercase PageNumRomanLC : Roman lowercase PageNumAlphaUC : Alphabetic uppercase PageNumAlphaLC : Alphabetic lowercase PageNumKanji : Kanji characters PageNumZenkaku : Zenkaku PageNumZenkakuUC : Zenkaku uppercase PageNumZenkakuLC : Zenkaku lowercase PageNumKanjiKazu : Kazu PageNumDaiji : Daiji
PagePrefix	String	Page prefix string. Obsolete for FM 6.0
PageSide	Integer	Page side to start the component document on: BkStartFromFile BkStartNextAvailable BkStartLeft BkStartRight
PageSuffix	String	Page suffix string Obsolete for FM 6.0
PgfNumbering	Integer	The component document's paragraph numbering: Obsolete for FM 6.0 BkContPgfNum BkRestartPgfNum
PgfNumComputeMethod	Integer	Paragraph Computing Method: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous file. NumRestart : Restart at 1. NumReadFromFile : Use the numbering properties from the document associated with this book component.
PrevComponentInBook	Object	Previous component in the book file (BookComponent)
PrintInclude	Integer	True if the component document is included in list of book files to be printed
TblFnCustNumString	String	Custom table footnote number text. <i>FM 6.0 or greater</i>
TblFnNumComputeMethod	Integer	Table Footnote Computing Method: <i>FM 6.0 or greater</i> NumRestart : Start at 1. NumReadFromFile : Use the numbering properties from the document associated with this book component.

Table 168: Book Component Properties (Page 5 of 5)

Property Name	Data Type	Description
TblFnNumStyle	Integer	Table Footnote numbering style: <i>FM 6.0 or greater</i> FnNumNumeric : Arabic FnNumRomanUC : Roman uppercase FnNumRomanLC : Roman lowercase FnNumAlphaUC : Alphabetic uppercase FnNumAlphaLC : Alphabetic lowercase FnNumKanji : Kanji characters FnNumZenkaku : Zenkaku FnNumZenkakuUC : Zenkaku uppercase FnNumZenkakuLC : Zenkaku lowercase FnNumKanjiKazu : Kazu FnNumDaiji : Daiji FnNumCustom : Custom numbering
Unique (<i>Read-Only</i>)	Integer	Unique ID of the book component.
VolNumComputeMethod	Integer	Volume Numbering Computing Method: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous volume. NumRestart : Use the number specified by VolumeNumber . NumSame : Use the same volume number as the previous file. NumReadFromFile : Use the numbering properties from the document associated with this book component.
VolumeNumber	Integer	Volume Number: Use this as the volume number when the VolNumComputeMethod property is set to NumRestart <i>FM 6.0 or greater</i>
VolumeNumStyle	Integer	The numbering style of the volume: <i>FM 6.0 or greater</i> NumStyleNumeric : Arabic. NumStyleRomanUC : Roman, uppercase. NumStyleRomanLC : Roman, lowercase. NumStyleAlphaUC : Alphabetic, uppercase. NumStyleAlphaLC : Alphabetic, lowercase. NumStyleKanji : Kanji. NumStyleZenkaku : Zenkaku. NumStyleZenkakuUC : Zenkaku, uppercase. NumStyleZenkakuLC : Zenkaku, lowercase. NumStyleKanjiKazu : Kazu. NumStyleDaiji : Daiji. NumStyleText : Text
VolumeNumText	String	Volume Number text, if the VolumeNumStyle property is set to NumStyleText . <i>FM 6.0 or greater</i>
XmlApplicationForBookComponent	String	The Xml Application name for the component. FM 9.0 or greater

Character formats

Information for a character format is stored in a CharFmt object. There is one CharFmt object for each character format in a FrameMaker document. You may retrieve a character format object by its name or by going through a list stored in the containing document object.

Object Name - CharFmt

Table 169: Character Format Properties (Page 1 of 3)

Property Name	Data Type	Description
Capitalization	Integer	The capitalization type: CapitalCaseNorm: normal capitalization (mixed uppercase and lowercase) CapitalCaseSmall: small caps CapitalCaseLower: lowercase letters only CapitalCaseUpper: uppercase letters only
ChangeBar	Integer	True if Change Bars are on.
CharTag	String	The character format's tag name.
Color	Object	Spot color (Color).
CombinedFontFamily	Object	Combined font definition (CombinedFontDefn)
FontAngle	Integer	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName (Read-Only)	String	The font's encoding
FontFamily	Integer	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).
FontPlatformName	String	Name that uniquely identifies a font on a specific platform. For combined fonts, this is the Asian font name.
FontPostScriptName	String	Name given to a font when it is sent to a PostScript printer. For combined fonts, this is the Asian font name.
WesternFontPlatformName	String	Name that uniquely identifies the Roman component of a combined font on a specific platform.
WesternFontPostScriptName	String	Name given to the Roman component of a combined font when it is sent to a PostScript printer.
FontSize	Metric	Font size (2 pt to 400 pt).
FontVariation	Integer	Font variation (specifies an index into the array of font variations provided by the session property FontVariationNames).
FontWeight	Integer	Font weight (specifies an index into the array of font weights provided by the session property FontWeightNames).

Table 169: Character Format Properties (Page 2 of 3)

Property Name	Data Type	Description
Language	Integer	Hyphenation and spell-checking language to use: LangBrazilian LangBritish LangCanadianFrench LangCatalan LangDanish LangDutch LangEnglish LangFinnish LangFrench LangGerman LangItalian LangNewDutch LangNewGerman LangNewSwissGerman LangNoLanguage LangNorwegian LangNynorsk LangPortuguese LangSpanish LangSwedish LangSwissGerman LangJapanese LangTraditionalChinese LangSimplifiedChinese LangKorean and FM 8.0 or greater LangGreek LangRussian LangCzech LangPolish LangHungarian LangTurkish LangSlovak LangSlovenian LangBulgarian LangCroatian LangEstonian LangLatvian LangLithuanian LangRomanian and FM 2015 or greater LangRomanian LangArabic LangHebrew
KernX	Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Name	String	The character format's name.
NextCharFmtInDoc (Read-Only)	Object	Next character format in document (CharFmt).
Outline	Integer	True if Outline is enabled (Macintosh only).
Overline	Integer	True if Overline is enabled.
PairKern	Integer	True if Pair Kern is enabled.
Position	Integer	Vertical position of character: PosNorm: Normal PosSub: Subscript PosSuper: Superscript
Shadow	Integer	True if Shadow is enabled (Macintosh only).
Stretch	Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	Integer	True if Strikethrough is enabled.
Underlining	Integer	Underlining type: CbNoUnderline CbSingleUnderline CbDoubleUnderline CbNumericUnderline
UseBkColor	Integer	True/False. FM 10 or greater
UseCapitalization	Integer	True if Capitalization property overrides default; False if As Is setting used.
UseChangeBar	Integer	True if ChangeBar property overrides default; False if As Is setting used.

Table 169: Character Format Properties (Page 3 of 3)

Property Name	Data Type	Description
UseColor	Integer	True if Color property overrides default; False if As Is setting used.
UseFontAngle	Integer	True if FontAngle overrides default; False if As Is setting used.
UseFontFamily	Integer	True if FontFamily overrides default; False if As Is setting used.
UseFontSize	Integer	True if FontSize overrides default; False if As Is setting used.
UseFontVariation	Integer	True if FontVariation overrides default; False if As Is setting used.
UseFontWeight	Integer	True if FontWeight overrides default; False if As Is setting used.
UseKernX	Integer	True if KernX overrides default; False if As Is setting used.
UseKernY	Integer	True if KernY overrides default; False if As Is setting used.
UseLanguage	Integer	True if Language property overrides default; False if As Is setting used.
UseOutline	Integer	True if Outline property overrides default; False if As Is setting used (Macintosh only).
UseOverline	Integer	True if Overline property overrides default; False if As Is setting used.
UsePairKern	Integer	True if PairKern property overrides default; False if As Is setting used.
UsePosition	Integer	True if Position overrides default; False if As Is setting used.
UseShadow	Integer	True if Shadow property overrides default; False if As Is setting used (Macintosh only).
UseStretch	Integer	True if Stretch property overrides default, False if As Is setting is used.
UseStrikethrough	Integer	True if Strikethrough property overrides default; False if As Is setting used.
UseTsume	Integer	True if Tsume property overrides default; False if As Is setting used.
UseUnderlining	Integer	True if Underlining property overrides default; False if As Is setting used.

Colors

Information for a character format is stored in a `Color` object. There is one `Color` object for each color in a FrameMaker document. You may retrieve a color object by its name or by going through a list stored in the containing document object.

Object Name - Color

Table 170: Color Properties (Page 1 of 2)

Property Name	Data Type	Description
Black	Metric	Percentage of black (metric 0% to 100%)
ColorOverprint	Integer	Overprint setting for the color: <code>ColorOverPrint</code> or <code>ColorKnockOut</code>
ColorTintPercent	Metric	Tint percentage (0 - 100) or the value <code>ColorNotTinted</code>

Table 170: Color Properties (Page 2 of 2)

Property Name	Data Type	Description
ColorPrintCtl	Integer	Type of color printing used in document: PrintSpot PrintProcess PrintNo
ColorTintPercent	Metric	The tint percentage
ColorViewCtl	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. Each two bit value can be one of the following: SepNormal SepNone SepWhite
Cyan	Metric	Percentage of cyan (metric 0% to 100%)
FamilyName	String	Color library name
InkName	String	Specifies the name of the color library pigment.
Magenta	Metric	Percentage of magenta (metric 0% to 100%)
Name	String	Name of color
NextColorInDoc (Read-Only)	Object	Next color in document (Color)
ReservedColor	Integer	Color names reserved by FrameMaker. Possible values: ColorNotReserved ColorCyan ColorMagenta ColorYellow ColorBlack ColorWhite ColorRed ColorGreen ColorBlue and for FM 8.0 or greater ColorDarkGrey ColorPaleGreen ColorForestGreen ColorRoyalBlue ColorMauve ColorLightSalmon ColorDarkYellow ColorSalmon
TintBaseColor	Object	Color from which the tint is derived (Color)
Yellow	Metric	Percentage of yellow (metric 0% to 100%)

Combined font definitions

The `CombinedFontDefn` object is used to represent each combined font in a document. `Doc` objects have a property to specify the first combined font in the document's list of combined fonts.

IMPORTANT: Combined fonts are stored with the document, not with the current session. The session property, `FontFamilyNames` returns a list of fonts available for the current session, but it does not include any combined fonts. To get a list of combined font definitions, use `FirstCombinedFontDefnInDoc` to get the first combined font definition in the document. From that you can build a list of combined font definitions using `NextCombinedFontDefnInDoc`.

Object name - CombinedFontDefn

CombinedFontDefn objects have the following properties.

Table 171: Combined Font Definition Properties

Property Name	Data Type	Description
AllowBaseFamilyBoldedAndObliqued	Integer	False, the bolding or italicizing the Western font will not automatically enbold or italicize the Asian text. True to allow this.
BaseFamily	Integer	Asian font family (specifies index into the arrays of font families provided by the session property, FontFamilyNames)
FontEncodingName (Read-Only)	String	Combined font's encoding, based on the BaseFamily
Name	String	Name of the combined font.
NextCombinedFontDefnInDoc (Read-Only)	Integer	Next combined font definition instance in the document (CombinedFontDefn)
UserString	String	A string to which clients can store private data.
WesternFamily	Integer	Western font family (specifies index into the arrays of font families provided by the session property, FontFamilyNames)
WesternShift	Metric	Baseline offset of Roman text expressed as a percentage of base font size (metric 1% to 1000%)
WesternSize	Metric	Scaling factor for Roman text expressed as a percentage of base font size (metric 1% to 1000%)

Condition Formats

A CondFmt object is used to represent each conditional text format in a document.

Doc objects also have properties that specify how all the condition formats in the document appear.

Object name - CondFmt

CondFmt objects have the following properties.

Table 172: Condition Format Properties (Page 1 of 2)

Property Name	Data Type	Description
BkColor	Integer	The background color. FM 10 or greater
CondFmtIsShown	Integer	True if the condition is shown. To hide text with a specified condition, set this property and the Doc property, ShowAll, to False.
Name	String	Name of the condition format.
NextCondFmtInDoc (Read-Only)	Object	Next condition format in document (CondFmt).
SepOverride	Object	Color separation format override (Color).

Table 172: Condition Format Properties (Page 2 of 2)

Property Name	Data Type	Description
StyleOverride	Integer	Style condition indicators for conditional text: CnChangeBar CnDoubleUnderline CnNoOverride CnOverline CnSingleUnderline CnStrikeThrough CnNumericUnderline CnNmricAndChngbar
UseBkColor	Integer	True if the background color is shown. False (Default) if not shown. FM 10 or greater
UseSepOverride	Integer	True if color specified by <code>SepOverride</code> is used instead of default.

Cross-references

A `XRef` object is used to represent a cross-reference instance and an `XRefFmt` object is used to represent a cross-reference format.

Object name - XRef

`XRef` objects have the following properties.

Table 173: Cross Reference Properties (Page 1 of 2)

Property Name	Data Type	Description
Element (<i>Read-Only</i>)	Object	If the cross-reference is in a FM+SGML document, the associated element 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.
NextXRefInDoc (<i>Read-Only</i>)	Object	Next cross-reference instance in document (<code>XRef</code>).
TextRange (<i>Read-Only</i>)	TextRange	Text range that the cross-reference instance encompasses.
XRefAltText	String	The alternate text of the cross-reference source marker. FM 9.0 or greater
XRefClientName	String	The name of the script that handles the client created <code>XRef</code> . FM 9.0 or greater
XRefClientType	String	The type of the script that handles the client created <code>XRef</code> . FM 9.0 or greater
XRefFmt	Object	The cross-reference's format object (<code>XrefFmt</code>).
XRefFile	String	The filename 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 ("").

Table 173: Cross Reference Properties (Page 2 of 2)

Property Name	Data Type	Description
XRefIsUnresolved (<i>Read-Only</i>)	Integer	True if the FrameMaker was unable to resolve the cross-reference the last time it updated cross-references. Note that this property is set only when the FrameMaker updates cross-references. Changes to the document, in and of themselves, do not affect this property.
XRefSrcElemNonUniqueId	String	The non unique id of the client created XRef. FM 9.0 or greater
XRefSrcText	String	The text of the cross-reference source marker.

Object name - XRefFmt

XRefFmt objects have the following properties.

Table 174: Cross Reference Format Properties

Property Name	Data Type	Description
Fmt	String	The cross-reference format (a string that specifies text and building blocks)
Name	String	The cross-reference format's name
NextXRefFmtInDoc (<i>Read-Only</i>)	Object	The next cross-reference format object(XRefFmt)

Documents

A Doc object is used to represent each open document in a FrameMaker session.

Object name - Doc

Doc objects have the following properties.

Table 175: Document Properties (Page 1 of 24)

Property Name	Data Type	Description
AcrobatBookmarkDisplayTags	Integer	True if Include Paragraph Tags in Bookmark Text is on (the paragraph tag is added before the paragraph text in each bookmark).
AutoChangeBars	Integer	True if Automatic Change Bars is enabled
BkColor	Object	Background color object(Color Object). FM 10 or greater .
BannerTextDisplay	String	Undocumented. FM 11 or greater
BooleanConditionExpression	String	
BooleanConditionExpressionTag	String	FM 10 or greater
BooleanConditionState	Integer	

Table 175: Document Properties (Page 2 of 24)

Property Name	Data Type	Description
BottomMargin	Metric	Bottom page margin
Capitalization	Integer	Type of capitalization: CapitalCaseNorm CapitalCaseSmall CapitalCaseLower CapitalCaseUpper
ChangeBar	Integer	True if Change Bars are enabled.
ChangeBarColor	Object	The spot color (Color Object)
ChangeBarDistance	Metric	Distance between change bar and text column
ChangeBarPosition	Integer	Position of change bars: CbColLeft : Left of Column CbColRight : Right of Column CbColNearest : Side Nearest to Page Edge CbColFurthest : Side Farthest from Page Edge
ChangeBarThickness	Metric	Width of change bars
ChapNumComputeMethod	Integer	Chapter numbering type: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous chapter. NumRestart : Use the value specified for ChapterNumber . NumSame : Use the same chapter number as for the previous file.
ChapterNumber	Integer	If the ChapNumComputeMethod property is NumRestart , then use this value as the chapter number. <i>FM 6.0 or greater</i>
ChapterNumStyle	Integer	The numbering style: <i>FM 6.0 or greater</i> NumStyleNumeric : Arabic. NumStyleRomanUC : Roman, uppercase. NumStyleRomanLC : Roman, lowercase. NumStyleAlphaUC : Alphabetic, uppercase. NumStyleAlphaLC : Alphabetic, lowercase. NumStyleKanji : Kanji. NumStyleZenkaku : Zenkaku. NumStyleZenkakuUC : Zenkaku, uppercase. NumStyleZenkakuLC : Zenkaku, lowercase. NumStyleKanjiKazu : Kazu. NumStyleDaiji : Daiji. NumStyleText : Text NumStyleIndicNumeric : <i>FM 2015 or greater</i> NumStyleFarsiNumeric : <i>FM 2015 or greater</i> NumStyleHebrewNumeric : <i>FM 2015 or greater</i> NumStyleAbjadNumeric : <i>FM 2015 or greater</i> NumStyleAlifbataNumeric : <i>FM 2015 or greater</i>
ChapterNumText	String	If the ChapterNumStyle is equal to NumStyleText , then this property is the chapter number string. <i>FM 6.0 or greater</i>

Table 175: Document Properties (Page 3 of 24)

Property Name	Data Type	Description
CharTag	String	Name of character format tag.
ColGap	Metric	Size of gap between text columns
Color	Object	Spot color (Color Object).
CombinedFontFamily	Object	Combined font definition (CombinedFontDefn)
CondFmtIsShown	Integer	True if condition is shown.
CurrentPage	Object	Current page (BodyPage, MasterPage, RefPage Object).
CurrentPgf	Object	The paragraph of the current insertion point. If there is no insertion point this will be NULL.
CustomElementList	StringList	List of tags to display when ElementCatalogDisplay is set to ElcatCustom
DefaultExclusions	StringList	List of exclusions inherited when this document is included in a structured book.
DefaultInclusions	StringList	List of inclusions inherited when this document is included in a structured book.
Dictionary	StringList	List of words to accept when spell- checking the document.
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
DitaMaxRefLevels	Integer	FM 2017 or greater Dita Max ref levels
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
DocAcrobatColumnArticleThreads	Integer	True if you want separate article threads for each column, False if you want separate article threads for each text frame. Note that DocAcrobatNoArticleThreads must be false.
DocAcrobatDefaultsChanged	Integer	True if the default heuristics for determining the paragraph level are disabled.
DocAcrobatElementList	StringList	List of element tags and context labels to include in bookmarks. This applies only to structured FrameMaker documents.
DocAcrobatElements	Integer	True if elements are used for bookmarks instead of paragraphs.
DocAcrobatNoArticleThreads	Integer	True if you do not want article threads in the resulting PDF.
DocCharFmtNameList (Read-Only)	StringList	List of character formats for this document.

Table 175: Document Properties (Page 4 of 24)

Property Name	Data Type	Description
DocCMSType	Integer	Values are: FM 2015 or greater CmsTypeNone CmsTypeDocumentum CmsTypeSharepoint CmsTypeDitaExchange
DocCondFmtNameList (Read-Only)	StringList	List of Condition formats for this document.
DocElementDefNameList (Read-Only)	StringList	List of Element defs for this document.
DocFluidFlow	Object	The Flow Object to set the fluid view. Set this to 0 to turn this off.
DocIsDoubleSided	Integer	True if two-sided page layout
DocIsHelp (Read-Only)	Int	True if the document is the FrameMaker's Help document.
DocIsModified (Read-Only)	Integer	True if document has been modified. While this property is read-only, you can modify a document without setting this property to True by setting Untouchable to True for the document before your client modifies it.
DocIsViewOnly	Integer	True if the document is a view-only document
DocMasterPageNameList (Read-Only)	StringList	List of master page names for this document.
DocMarkerTypeNameList (Read-Only)	StringList	List of marker type names for this document.
DocOpenType (Read-Only)	Integer	Type of document the file was opened as: DocTypeBinary: Frame binary document DocTypeText: ASCII text document DocTypeMif: MIF document DocTypeFilter: a filtered document
DocPgFmtNameList (Read-Only)	StringList	List of paragraph formats for this document.
DocRefPageNameList (Read-Only)	StringList	List of reference page names for this document.
DocSaveType (Read-Only)	Integer	Type of document the file is saved as: DocTypeBinary: Frame binary document DocTypeText: ASCII text document DocTypeMif: MIF document DocTypeFilter: a filtered document
DocTblFmtNameList (Read-Only)	StringList	List of table formats for this document.
DocXRefFmtNameList (Read-Only)	StringList	List of cross reference formats for this document.

Table 175: Document Properties (Page 5 of 24)

Property Name	Data Type	Description
<code>DocVarFmtNameList</code> (Read-Only)	StringList	List of variable formats for this document.
<code>DocVarFmtNameSystemList</code> (Read-Only)	StringList	List of system variable formats for this document.
<code>DocVarFmtNameUserList</code> (Read-Only)	StringList	List of user variable formats for this document.
<code>DontUpdateTextInsets</code>	Integer	True if FrameMaker doesn't automatically update text insets when it opens the document.
<code>DontUpdateXRefs</code>	Integer	True if FrameMaker doesn't automatically update cross-references when it opens or prints the document.
<code>DownloadFonts</code>	Integer	The number of the entry in the Download Fonts popup menu of the Unix Print dialog box. The first popup entry is number 1.
<code>ElementBoundaryDisplay</code>	Integer	Element boundary display options: ElemDispNone - don't display any boundaries ElemDispBrackets - display bracketed boundaries ElemDispTags - display the element tags
<code>ElementCatalog</code>	EltCat	List of elements in the catalog
<code>ElementCatalogDisplay</code>	Integer	Catalog display options: ElcatStrict - valid children for working start to finish ElcatLoose - valid children for working in any order ElcatChildren - children allowed anywhere in parent ElcatAll - all elements ElcatCustom - the list of tags specified by the CustomElementList property
<code>ElementSelection</code>	ElementRange	The currently selected element range in the document.
<code>EqnIntegralSizeLarge</code>	Metric	Point size of integral symbol in large equations (2 pt to 400 pt)
<code>EqnIntegralSizeMed</code>	Metric	Point size of integral symbol in medium equations (2 pt to 400 pt)
<code>EqnIntegralSizeSmall</code>	Metric	Point size of integral symbol in small equations (2 pt to 400 pt)
<code>EqnLevel1SizeLarge</code>	Metric	Point size of level 1 expression in large equations (2 pt to 400 pt)
<code>EqnLevel1SizeMed</code>	Metric	Point size of level 1 expression in medium equations (2 pt to 400 pt)
<code>EqnLevel1SizeSmall</code>	Metric	Point size of level 1 expression in small equations (2 pt to 400 pt)
<code>EqnLevel2SizeLarge</code>	Metric	Point size of level 2 expression in large equations (2 pt to 400 pt)
<code>EqnLevel2SizeMed</code>	Metric	Point size of level 2 expression in medium equations (2 pt to 400 pt)
<code>EqnLevel2SizeSmall</code>	Metric	Point size of level 2 expression in small equations (2 pt to 400 pt)
<code>EqnLevel3SizeLarge</code>	Metric	Point size of level 3 expression in large equations (2 pt to 400 pt)
<code>EqnLevel3SizeMed</code>	Metric	Point size of level 3 expression in medium equations (2 pt to 400 pt)
<code>EqnLevel3SizeSmall</code>	Metric	Point size of level 3 expression in small equations (2 pt to 400 pt)
<code>EqnSigmaSizeLarge</code>	Metric	Point size of sigma symbol in large equations (2 pt to 400 pt)
<code>EqnSigmaSizeMed</code>	Metric	Point size of sigma symbol in medium equations (2 pt to 400 pt)

Table 175: Document Properties (Page 6 of 24)

Property Name	Data Type	Description
<code>EqnSigmaSizeSmall</code>	Metric	Point size of sigma symbol in small equations (2 pt to 400 pt)
<code>FileExtensionOverride</code>	String	The file extension to use when saving this document as XML. Typically this is used to save XHTML with a <code>.htm</code> extension rather than <code>.xml</code> . This setting should be set in the structure application for this document's DOCTYPE, and that setting may be a more reliable source for this value. (FM 7.0 or greater)
<code>FileInfoPacket</code>	String	The XMP Metadata. See "XMP Metadata" on page 194. (FM 7.0 or greater)
<code>FirstAttrCondExprInDoc</code> (Read-Only)	Object	First <code>AttrCondExpr</code> Object in the document (<code>AttrCondExpr</code> Object)
<code>FirstBodyPageInDoc</code> (Read-Only)	Object	First body page in the document (<code>BodyPage</code> Object)
<code>FirstCharFmtInDoc</code> (Read-Only)	Object	First character tag in the list of the document's character tags (<code>CharFmt</code> Object)
<code>FirstColorInDoc</code> (Read-Only)	Object	First color in the list of document's colors (<code>Color</code> Object)
<code>FirstCombinedFontDefnInDoc</code> (Read-Only)	Object	First combined font definition in the list of the document's combined font definitions (<code>CombinedFontDefn</code> Object)
<code>FirstCondFmtInDoc</code> (Read-Only)	Object	First condition tag in the list of the document's condition tags (<code>CondFmt</code> Object)
<code>FirstDITAConrefElementInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstDITALinkElementInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstDITATopicrefElementInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstDITATopicsetrefElementInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstDITAXrefElementInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstElementDefInDoc</code> (Read-Only)	Object	First element definition in the document (<code>ElementDef</code> Object)
<code>FirstFlowInDoc</code> (Read-Only)	Object	First flow in the list of the document's flows (<code>Flow</code> Object)
<code>FirstFmtChangeListInDoc</code> (Read-Only)	Object	First format change list in the document (<code>FmtChangeList</code> Object)
<code>FirstFnInDoc</code> (Read-Only)	Object	First footnote in the list of the documents footnotes (<code>Fn</code> Object)
<code>FirstGraphicInDoc</code> (Read-Only)	Object	First graphic object in the list of the document's graphic objects (<code>GraphicObject</code> Object)
<code>FirstGraphicsFmtInDoc</code> (Read-Only)	Object	Undocumented. FM 11 or greater
<code>FirstInlineComponentInDoc</code> (Read-Only)	Object	First inline component (<code>InlineComponent</code> Object). FM 2015 or greater

Table 175: Document Properties (Page 7 of 24)

Property Name	Data Type	Description
FirstMarkerInDoc (<i>Read-Only</i>)	Object	First marker in the list of the document's markers (Marker Object)
FirstMarkerTypeInDoc (<i>Read-Only</i>)	Object	First marker type in the list of the document's marker types (MarkerType Object).
FirstMasterPageInDoc (<i>Read-Only</i>)	Object	First master page in the document (MasterPage Object)
FirstPageNum	Integer	Page number of first page
FirstPageVerso	Integer	False for right first page; True for left first page
FirstPgFmtInDoc (<i>Read-Only</i>)	Object	First paragraph tag in the list of the document's paragraph tags (PgFmt Object)
FirstPgInDoc (<i>Read-Only</i>)	Object	First paragraph (Pgf Object) in the list of the document's paragraphs
FirstPgInMainFlow (<i>Read-Only</i>)	Object	First paragraph (Pgf Object) in the main flow (MainFlowInDoc) of the document.
FirstRefPageInDoc (<i>Read-Only</i>)	Object	First reference page in the document (RefPage Object)
FirstRubiInDoc (<i>Read-Only</i>)	Object	First rubi composite in the list of the document's rubi composites (Rubi Object)
FirstRulingFmtInDoc (<i>Read-Only</i>)	Object	First ruling format in the list of the document's ruling formats (RulingFmt Object)
FirstSelectedGraphicInDoc (<i>Read-Only</i>)	Object	First selected graphic object in the document's list of selected graphic objects (Graphic Object).
FirstSelectedTiInDoc (<i>Read-Only</i>)	Object	First selected text inset in the list of selected text insets in the document (TiApiClient, TiText, TiTextTable, or TiFlow Object)
FirstTblFmtInDoc (<i>Read-Only</i>)	Object	First table format in the list of the document's table formats (TblFmt Object)
FirstTblInDoc (<i>Read-Only</i>)	Object	First table in the list of the document's tables (Tbl Object)
FirstTiInDoc (<i>Read-Only</i>)	Object	First text inset in the list of the document's text insets (TiApiClient, TiText, TiTextTable, or TiFlow Object)
FirstVarFmtInDoc (<i>Read-Only</i>)	Object	First variable format Object in the list of the document's variable formats (VarFmt)
FirstVarInDoc (<i>Read-Only</i>)	Object	First variable Object in the list of the document's variables (Var)
FirstXRefFmtInDoc (<i>Read-Only</i>)	Object	First cross-reference format Object in the list of the document's cross-reference formats (XRefFmt)
FirstXRefInDoc (<i>Read-Only</i>)	Object	First cross-reference Object in the list of the document's cross-references (XRef)
FnCustNumString	String	Characters for custom document footnote numbers
FnFirstNum	Integer	First document footnote number
FnFmt	String	Paragraph tag of footnote
FnHeightPerCol	Metric	Maximum height allowed for document footnotes (36 pt to 32767 pt)

Table 175: Document Properties (Page 8 of 24)

Property Name	Data Type	Description
FnInstancePosition	Integer	Placement of document footnote number in footnote: FnPosSuper: Superscript FnPosBaseline: Baseline FnPosSub: Subscript
FnInstancePrefix	String	Prefix to appear before document footnote number in footnote
FnInstanceSuffix	String	Suffix to appear after document footnote number in footnote
FnNumberingPerPage	Integer	True if document footnote numbering is by page, rather than by flow
FnNumComputeMethod	Integer	Footnote Computing Method: <i>FM 6.0 or greater</i> NumContinue: Continue the numbering from the previous file. NumRestart: Use the number specified by FnFirstNum. NumPerPage: Restart numbering on each page.
FnNumStyle	Integer	Document footnote numbering style: FnNumNumeric: Arabic FnNumRomanUC: Roman uppercase FnNumRomanLC: Roman lowercase FnNumAlphaUC: Alphabetic uppercase FnNumAlphaLC: Alphabetic lowercase FnNumKanji: Kanji characters FnNumZenkaku: Zenkaku FnNumZenkakuUC: Zenkaku uppercase FnNumZenkakuLC: Zenkaku lowercase FnNumKanjiKazu: Kazu FnNumDaiji: Daiji FnNumCustom: Custom numbering FnNumIndicNumeric: FM 2015 or greater FnNumFarsiNumeric: FM 2015 or greater FnNumHebrewNumeric: FM 2015 or greater FnNumAbjadNumeric: FM 2015 or greater FnNumAlifbataNumeric: FM 2015 or greater FnNumFarsiAlpha: FM 2017 or greater FnNumHebrewAlpha: FM 2017 or greater FnNumThaiAlpha: FM 2017 or greater FnNumThaiNumeric: FM 2017 or greater
FnRefPosition	Integer	Position of footnote reference in document text: FnPosSuper: Superscript FnPosBaseline: Baseline FnPosSub: Subscript
FnRefPrefix	String	Prefix to appear before number in document text
FnRefSuffix	String	Suffix to appear after number in document text

Table 175: Document Properties (Page 9 of 24)

Property Name	Data Type	Description
FontAngle	Integer	Font angle (specifies an index into the array of font angles provided by the session property, <code>FontAngleNames</code>).
FontEncodingName (Read-Only)	String	The font's encoding
FontFamily	Integer	Font family (specifies an index into the array of font families provided by the session property, <code>FontFamilyNames</code>).
FontPlatformName	String	Name that uniquely identifies a font on a specific platform.
FontPostScriptName	String	Name given to a font when it is sent to a PostScript printer.
FontSize	Metric	Font size (2 pt to 400 pt).
FontVariation	Integer	Font variation (specifies an index into the array of font variations provided by the session property <code>FontVariationNames</code>).
FontWeight	Integer	Font weight (specifies an index into the array of font weights provided by the session property <code>FontWeightNames</code>).
FormatOverride	Integer	Specifies whether there are format overrides at the current insertion point. If the insertion point is in a text range that has a character format applied to it, <code>FormatOverride</code> is <code>True</code> if (and only if) the text formatting at the insertion point overrides the character format. If the insertion point is in a text range that has does not have a character format applied to it, <code>FormatOverride</code> is <code>True</code> if (and only if) the paragraph containing the insertion point has formatting that overrides the Paragraph Catalog format.
Functions	String	Character format tag of equation font to apply to Math Functions
GenerateAcrobatInfo	Integer	True if Generate Adobe Acrobat Data is on. To generate Adobe Acrobat data, you must set other document print properties as follows PrintToFile: True PrintThumbnails: False PrintSeps: False PrintBlankPages: True PrintLastSheetFirst: False PrintNumCopies: 1 PrintOddPages: True PrintEvenPages: True PrintScale: 100%
HiddenPage (Read-Only)	Object	Hidden page (<code>HiddenPage</code>) Object
HypertextCommandText	String	The hypertext command to parse. Setting this value executes the parser. If <code>HypertextDoValidate</code> is <code>True</code> , the command will be parsed and validated.
HypertextDoValidate	BoolT	True if the next hypertext string sent to <code>HypertextCommandText</code> will be validated
HypertextParseBadParam (Read-Only)	Integer	If there was a parse error, an index into the <code>HypertextParsedArgs</code> string list.
HypertextParsedArgs (Read-Only)	StringList	The value of <code>HypertextCommand</code> , parsed into individual tokens
HypertextParsedClientName (Read-Only)	String	For message commands, the name of the script to receive the message.

Table 175: Document Properties (Page 10 of 24)

Property Name	Data Type	Description
HypertextParsedCmdCode (Read-Only)	Integer	The FrameMaker hypertext command in <code>HypertextCommandText</code> , as determined by the parser. See “Hypertext Command Code Table” on page 196.
HypertextParsedCmdDest (Read-Only)	Integer	For link commands, the destination type in <code>HypertextCommandText</code> , as determined by the parser. See “Link Destination Object Type Table” on page 198.
HypertextParsedCmdDestObjID (Read-Only)	Integer	For links to objects, the object in the target document.
HypertextParsedCmdDestObjType (Read-Only)	Integer	For links to objects, the type of the object in the target document.
HypertextParsedCmdMatrixColumns (Read-Only)	Integer	If <code>HypertextParsedCmdCode</code> is <code>CmdMatrix</code> , the number of columns in the matrix.
HypertextParsedCmdMatrixRows (Read-Only)	Integer	If <code>HypertextParsedCmdCode</code> is <code>CmdMatrix</code> , the number of rows in the matrix.
HypertextParsedDIFFileName (Read-Only)	String	For links to external files, the absolute path to the target file, expressed in platform independent syntax.
HypertextParsedFlowName (Read-Only)	String	For popup and matrix commands, the name of the flow (on a reference page) that contains the popup or matrix list of commands.
HypertextParsedLinkName (Read-Only)	String	For links to named targets, either the value of a newlink command, or a keyword such as <code>FirstPage</code> or <code>LastPage</code> .
HypertextParsedMessage (Read-Only)	String	If <code>HypertextParsedCmdCode</code> is <code>CmdAlert</code> , <code>CmdAlertTitle</code> , or <code>CmdMessage</code> , the specified message for the hypertext command.
HypertextParsedPageName (Read-Only)	String	For links to pages, the page number.
HypertextParsedTitle (Read-Only)	String	If <code>HypertextParsedCmdCode</code> is <code>CmdAlertTitle</code> , the specified title for the alert box.
HypertextParseErr (Read-Only)	Integer	Non-zero if there was a parse error.
HypertextParseErrMsg (Read-Only)	String	The message FrameMaker generates for a parse error
HypertextValidateErr (Read-Only)	Integer	Non-zero if <code>HypertextDoValidate</code> was true and there was a validation error.
InCond	IntList	List of condition codes applicable at the this text location. This is a list of integers. You must use the New Object command to convert one of these integers into a condition object.
IsIconified	Integer	True if the document window is iconified (UNIX and Windows only).
IsInFront	Integer	True if the document window is in front of other windows in the FrameMaker session.
IsOnScreen	Integer	True if document is visible on the screen.
IsOnScreen	Integer	True if document is visible on the screen.
KernX	Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric -100% to 1000%). A positive value moves a character right and a negative value moves a character left.

Table 175: Document Properties (Page 11 of 24)

Property Name	Data Type	Description
KernY	Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
KeyCatalog (Read-Only)	Object	Undocumented. FM 11 or greater
KeyCatalogType	Integer	Undocumented. FM 11 or greater
Label	String	The title in the document window title bar.
LastBodyPageInDoc (Read-Only)	Object	Last body page in the document (BodyPage)Object
LastMasterPageInDoc (Read-Only)	Object	Last master page (MasterPage) Object
LastPgfInMainFlow (Read-Only)	Object	First paragraph (Pgf Object) in the main flow (MainFlowInDoc) of the document.
LastRefPageInDoc (Read-Only)	Object	Last reference page in the document (RefPage) Object
LeftMargin	Metric	Left page margin
LeftMasterPage (Read-Only)	Object	Left master page (MasterPage) Object
LineBreakAfter	String	Characters at which it is permissible to break lines
LineNumDistance	Metric	Undocumented. FM 11 or greater
LineNumRestart	Integer	Undocumented. FM 11 or greater
LineNumShow	Integer	Undocumented. FM 11 or greater
MainFlowInDoc (Read-Only)	Object	Main flow (Flow) Object
MarkerTypeNames (Read-Only)	StringList	List of markertype names
MaxBottomMargin	Metric	Maximum bottom margin allowed in the document.
MaxFirstIndent	Metric	Maximum first indent allowed in the document.
MaxFontSize	Metric	Maximum font size allowed in the document.
MaxLeading	Metric	Maximum leading allowed in the document.
MaxLeftIndent	Metric	Maximum left indent allowed in the document.
MaxLeftMargin	Metric	Maximum left margin allowed in the document.
MaxRightIndent	Metric	Maximum right indent allowed in the document.
MaxRightMargin	Metric	Maximum right margin allowed in the document.
MaxSpaceAbove	Metric	Maximum space above paragraph allowed in the document.
MaxSpaceBelow	Metric	Maximum space below paragraph allowed in the document.
MaxSpread	Metric	FrameMaker 5.1x only. FrameMaker 5.5 use the "tracking" property.
MaxStretch	Metric	Maximum character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).

Table 175: Document Properties (Page 12 of 24)

Property Name	Data Type	Description
MaxTabPosition	Metric	Maximum tab position allowed in the document.
MaxTopMargin	Metric	Maximum top margin allowed in the document.
MenuBar	Object	The document's menu bar object (Menu)
MinBottomMargin	Metric	Minimum bottom margin allowed in the document.
MinFirstIndent	Metric	Minimum first indent allowed in the document.
MinFontSize	Metric	Minimum font size allowed in the document.
MinLeading	Metric	Minimum leading allowed in the document.
MinLeftIndent	Metric	Minimum left indent allowed in the document.
MinLeftMargin	Metric	Minimum left margin allowed in the document.
MinRightIndent	Metric	Minimum right indent allowed in the document.
MinRightMargin	Metric	Minimum right margin allowed in the document.
MinSpaceAbove	Metric	Minimum space above paragraph allowed in the document.
MinSpaceBelow	Metric	Minimum space below paragraph allowed in the document.
MinSpread	Metric	FrameMaker 5.1x only. FrameMaker 5.5 use the "tracking" property.
MinStretch	Metric	Minimum character stretch (set width) expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
MinTabPosition	Metric	Minimum tab position allowed in the document.
MinTopMargin	Metric	Minimum top margin allowed in the document.
Name (Read-Only)	String	Filename of the document.
NarrowRubiSpaceForJapanese	Integer	Allowable values are: Wide Narrow Proportional
NarrowRubiSpaceForOther	Integer	Allowable values are: Wide Narrow Proportional
NewElemAttrDisplay	Integer	Specifies attribute display properties for new elements: AttrDispNone: don't display attributes AttrDispReqSpec: display required and specified attributes AttrDispAll: display all attributes
NewElemAttrEditing	Integer	Specifies when the Edit Attributes dialog box appears for new elements: AttrEditNone AttrEditRequired AttrEditAlways
NextOpenDocInSession (Read-Only)	Object	Next open document Object in the list of open documents in the session (Doc)
Numbers	String	Character format tag of equation font to apply to Math Numbers
NumCols (Read-Only)	Integer	Number of columns
Outline	Integer	True if Outline style is enabled (Macintosh only).
Overline	Integer	True if Overline style is enabled.

Table 175: Document Properties (Page 13 of 24)

Property Name	Data Type	Description
PageHeight	Metric	Height of the document's pages. Setting this property automatically sets the <code>PageHeight</code> property of all of the document's body pages.
PageNumComputeMethod	Integer	Page Computing Method: <i>FM 6.0 or greater</i> NumContinue: Continue the numbering from the previous file. NumRestart: Use the number specified by <code>FirstPageNum</code> .
PageNumStyle	Integer	Page numbering style: <code>PageNumNumeric</code> : Arabic <code>PageNumRomanUC</code> : Roman uppercase <code>PageNumRomanLC</code> : Roman lowercase <code>PageNumAlphaUC</code> : Alphabetic uppercase <code>PageNumAlphaLC</code> : Alphabetic lowercase <code>PageNumAlphaLC</code> : Kanji characters <code>PageNumZenkaku</code> : Zenkaku <code>PageNumZenkakuUC</code> : Zenkaku uppercase <code>PageNumZenkakuLC</code> : Zenkaku lowercase <code>PageNumKanjiKazu</code> : Kazu <code>PageNumDaiji</code> : Daiji PageNumIndic: FM 2015 or greater PageNumFarsi: FM 2015 or greater PageNumHebrew: FM 2015 or greater PageNumAbjad: FM 2015 or greater PageNumAlifbata: FM 2015 or greater PageNumFarsiAlpha: FM 2017 or greater PageNumHebrewAlpha: FM 2017 or greater PageNumThaiAlpha: FM 2017 or greater
PageRounding	Integer	How to round pages: <code>PrDelEmpty</code> : Delete Empty Pages <code>PrKeepNumEven</code> : Make Page Count Even <code>PrKeepNumOdd</code> : Make Page Count Odd <code>PrDontChange</code> : Don't Change Page Count
PageWidth	Metric	Width of the document's pages. Setting this property automatically sets the <code>PageWidth</code> property of all of the document's body pages.
PairKern	Integer	True if Pair Kern is enabled.
PDFAllNamedDestinations	Integer	True if PDFs generated from this document will have named destinations for every paragraph and SGML element. <i>FM 6.0 or greater</i>
PDFBookMark	Integer	True if Frame will generate bookmarks when saving to PDF. <i>FM 6.0 or greater</i>

Table 175: Document Properties (Page 14 of 24)

Property Name	Data Type	Description
PDFBookmarksOpenLevel	Integer	The level of bookmarks to have expanded when Acrobat opens the generated PDF document. This can be any integer or one of the following values: PDFBookmarksOpenDefaultLevel PDFBookmarksOpenAllLevels PDFBookmarksOpenNoneLevel FM 7.0 or greater
PDFConvertCMYKtoRGB	Integer	When True this corresponds with setting Convert CMYK colors to RGB in the Save As PDF dialog box. Only has effect on Macintosh. FM 7.0 or greater
PDFDestsMarked	Integer	True if the document has elements marked with the MarkedForNamedDestination. <i>FM 6.0 or greater</i>
PDFDistillerAbsent (Read-Only)	Integer	Set to 1 if there is no Acrobat Distiller available. .FM 7.0 or greater
PDFDocInfo	StringList	A list of strings containing the values to be set for the PDF dictionary. <i>FM 6.0 or greater</i>
PDFEndPage	String	The last page of the printing page range, in the FrameMaker numbering style. .FM 7.0 or greater
PDFGenerateForReview	Integer	Pdf Settings Review. FM 9.0 or greater
PDFJobOption	String	The name of the distiller Job options. If the specified name does not exist in the Distiller Job Options list then the first Distiller Job Option in the list is used. .FM 7.0 or greater
PDFJobOptionsAbsent (Read-Only)	Integer	Set to 1 if there is no PDF Job Options available. .FM 7.0 or greater
PDFOpenPage	String	The PDF page number, in the FrameMaker numbering style, at which Acrobat opens the generated PDF document. .FM 7.0 or greater
PDFPageHeight	Metric	The Page height for the generated PDF document. .FM 7.0 or greater
PDFPageWidth	Metric	The Page width for the generated PDF document. .FM 7.0 or greater
PDFPrintPageRange	Integer	True for generating PDF for the specified page range. False otherwise. .FM 7.0 or greater
PDFRegistrationMarks	Integer	Registration marks for the generated PDF document. Use one of the following values: PDFRegistrationMarksNone PDFRegistrationMarksWestern PDFRegistrationMarksTombo .FM 7.0 or greater
PDFSeparateFiles	Integer	True for generating a separate PDF document for each document in the book. False otherwise. .FM 7.0 or greater

Table 175: Document Properties (Page 15 of 24)

Property Name	Data Type	Description
PDFStartPage	String	The first page of the printing page range, in the FrameMaker numbering style. .FM 7.0 or greater
PDFStructure	Integer	True if Frame will generate structured PDF when saving to PDF. (Set the PDFStructureLevel property of the PgfFmt object.) <i>FM 6.0 or greater</i>
PDFViewPDF	Integer	True if Frame will generate structured PDF when saving to PDF. (Set the PDFStructureLevel property of the PgfFmt object.) <i>FM 6.0 or greater</i>
PDFZoomFactor	Metric	When PDFZoomType is set to PDFZoomNone, this is the zoom percentage of the PDF document. .FM 7.0 or greater
PDFZoomType	Integer	The PDF zoom setting with which the Acrobat opens the generate PDF document. Can be one of the following values: PDFZoomDefault PDFZoomPage PDFZoomWidth PDFZoomHeight PDFZoomNone .FM 7.0 or greater
PgfCell	Object	If the current insertion point is located inside a table cell, this will be that cell object, otherwise it is NULL.
PgfNumComputeMethod	Integer	Paragraph Computing Method: <i>FM 6.0 or greater</i> NumContinue: Continue the numbering from the previous file. NumRestart: Restart at 1.

Table 175: Document Properties (Page 16 of 24)

Property Name	Data Type	Description
PointPageNumStyle	Integer	Point page numbering style: PointPageNumNumeric: Arabic PointPageNumRomanUC: Roman uppercase PointPageNumRomanLC: Roman lowercase PointPageNumAlphaUC: Alphabetic uppercase PointPageNumAlphaLC: Alphabetic lowercase PointPageNumKanji: Kanji characters PointPageNumZenkaku: Zenkaku PointPageNumZenkakuUC: Zenkaku uppercase PointPageNumZenkakuLC: Zenkaku lowercase PointPageNumKanjiKazu: Kazu PointPageNumDaiji: Daiji PointPageNumIndic: FM 2015 or greater PointPageNumFarsi: FM 2015 or greater PointPageNumHebrew: FM 2015 or greater PointPageNumAbjad: FM 2015 or greater PointPageNumAlifbata: FM 2015 or greater PointPageNumFarsiAlpha: FM 2017 or greater PointPageNumHebrewAlpha: FM 2017 or greater PointPageNumThai: FM 2017 or greater PointPageNumThaiAlpha: FM 2017 or greater
Position	Integer	Text position relative to baseline of text: PosNorm: Normal PosSub: Subscript PosSuper: Superscript
PreviewState	Integer	Preview state. FM 8.0 or greater Possible values are PreviewOffTrackChange PreviewOnOriginal PreviewOnFinal
PrintBlankPages	Integer	True if PageRounding allows empty page at end of document.
PrintCollated	Integer	True if Collate is enabled.
PrintCols	Integer	If PrintThumbnails is True, the number of columns to print.
PrintDitaFilename	String	The Dita Filename.
PrintEmulsion	Integer	Direction of print emulsion: EmulUp: Emulsion side up EmulDown: Emulsion side down
PrintEndPage	Integer	Number of last page to print. Note that the value of DocFluidFlow must be false; you can't print a range of pages when a document is in fluid view.
PrintEndPageName	String	Page number string of last page to print. Use this when the pages are numbered with a style other than PageNumNumeric. Note that the value of DocFluidFlow must be false; you can't print a range of pages when a document is in fluid view.

Table 175: Document Properties (Page 17 of 24)

Property Name	Data Type	Description
PrintEndPoint	Integer	Number of last point page to print.
PrinterName	String	Name of printer. Setting <code>PrinterName</code> on Windows has no effect. When you set <code>PrinterName</code> , you can set the printer to the default printer by specifying <code>ZERO</code> .
PrintEvenPages	Integer	<code>True</code> if Print Even-Numbered Pages is enabled.
PrintFileName	String	Filename of file to print to. When you set <code>PrintFileName</code> , you can set the filename to the default filename by specifying <code>ZERO</code> . Setting this property on the Macintosh has no effect.
PrintImaging	Integer	Type of print imaging: ImgPositive ImgNegative
PrintLastSheetFirst	Integer	<code>True</code> if Last Sheet First is enabled.
PrintLowRes	Integer	<code>True</code> if Low-Resolution is enabled.
PrintManualFeed	Integer	<code>True</code> if Manual Feed is enabled.
PrintNumCopies	Integer	Number of copies to print.
PrintOddPages	Integer	<code>True</code> if Odd-Numbered Pages is enabled.
PrintPaperHeight	Metric	Height of paper.
PrintPaperWidth	Metric	Width of paper.
PrintRegistrationMarks	Integer	<code>True</code> if Registration Marks is enabled.
PrintRows	Integer	If <code>PrintThumbnails</code> is <code>True</code> , the number of rows to print.
PrintScale	Integer	Scale factor.
PrintScope	Integer	Pages to print. Note that the value of <code>DocFluidFlow</code> must be <code>false</code> ; you can't print a range of pages when a document is in fluid view. <code>PrAll</code> : Print all pages <code>PrRange</code> : Print a range of pages
PrintSeeps	Integer	<code>True</code> if Print Separations is enabled.
PrintStartPage	Integer	Number of the first page to print. Note that the value of <code>DocFluidFlow</code> must be <code>false</code> ; you can't print a range of pages when a document is in fluid view.
PrintStartPageName	String	Page number string of first page to print. Use this when the pages are numbered with a style other than <code>PageNumNumeric</code> . Note that the value of <code>DocFluidFlow</code> must be <code>false</code> ; you can't print a range of pages when a document is in fluid view.
PrintStartPoint	Integer	Number of first point page to print.
PrintThumbnails	Integer	<code>True</code> if Print Thumbnails is enabled.
PrintToFile	Integer	<code>True</code> if Print Only to File is enabled. Setting this property on the Macintosh has no effect.
PrintTomboMarks	BoolT	<code>True</code> if registration marks are enabled, and set to Tombo. When printing Tombo Marks, you must also set <code>PrintRegistrationMarks</code> to <code>True</code> .
ReviewerName	String	Name of the reviewer. FM 10 or greater

Table 175: Document Properties (Page 18 of 24)

Property Name	Data Type	Description
ReviewerNameList	StringList	Names of the reviewers. FM 10 or greater
RightMargin	Metric	Right page margin
RightMasterPage (<i>Read-Only</i>)	Object	Right master page (MasterPage) Object
RubiAlignAtLineBounds	Integer	TRUE if rubi and oyamoji text should be aligned at line boundaries
RubiFixedSize	Metric	Fixed size for all rubi text (metric 2pts to 400pts). If this property and the RubiSize property both have values, the most recently set property value is used.
RubiOverhang	Integer	TRUE if rubi is allowed to overhang
RubiSize	Metric	Scaling factor for rubi text expressed as percentage of the current font size (metric 1% to 1000%). If this property and the RubiFixedSize property both have values, the most recently set property value is used.
ScreenHeight	Integer	Height of the document window in pixels.
ScreenWidth	Integer	Width of the document window in pixels.
ScreenX	Integer	The offset of the document window in pixels from the left side of the screen (or the top of the FrameMaker application window on Windows). If you set a value that would result in the document window being off the screen, that value is ignored and the old value is retained.
ScreenY	Integer	The offset of the document window in pixels from the top of the screen (or the top of the FrameMaker application window on Windows). If you set a value that would result in the document window being off the screen, that value is ignored and the old value is retained.
SelectedTbl (<i>Read-Only</i>)	Object	If any table cells are selected or if the insertion point is located inside a table cell, the table Object containing them (Tbl, NULL otherwise).
SelectedTblCell (<i>Read-Only</i>)	Object	If any table cells are selected, the first Table Cell Object (Cell), NULL otherwise
SeparateInclusions	Integer	True if inclusions are listed separately in the Element Catalog.
SepOverride	Object	Custom color separation override (Color Object).
SgmlApplication	String	The name of the SGML application associated with the document. (FM 5.5.6 or FM 6.0)
Shadow	Integer	True if Shadow style is enabled (Macintosh only).
ShowAll	Integer	True if all conditions are displayed
ShowCondIndicators	Integer	True if condition indicators (format overrides) are displayed
ShowElementDescriptiveNames	Integer	True/False. FM 10 or greater
SkipBlankSeps	Integer	True if Skip Blank Separations is enabled (don't print blank color separations).
SmallCapsSize	Metric	Scaling factor for small caps expressed as percentage of current font size (metric 1% to 1000%)
SmallCapsStretch	Metric	Character stretch (set width) for small caps expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
SmartQuotes	Integer	True if Smart Quotes is enabled

Table 175: Document Properties (Page 19 of 24)

Property Name	Data Type	Description
SmartSpaces	Integer	True if Smart Spaces is enabled
SnapAngle	Metric	Angle of rotation for Snap Rotate.
SnapGridUnits	Metric	Units for Snap Grid Spacing (0 to 32768 pt).
SpecifiedKeyCatalog	Object	Undocumented. FM 11 or greater
SpotColorView	Integer	Spot color separation view (0 to 6). 0 specifies View 1, 1 specifies View 2, and so on.
SpotColorView	Integer	The Color View of the document. The values are 0 through 5 which correspond to the color view 1 through 6.
Spread	Metric	FrameMaker 5.1x only. FrameMaker 5.5 use the "tracking" property.
StatusLine	String	String that appears in the document status bar. Note that this property always returns an empty string; it is effectively write-only If you set <code>StatusLine</code> to a string other than an empty string (' '), the string will remain in the status bar until you reset it. To reset <code>StatusLine</code> so the FrameMaker automatically updates the status line with normal status information, set it to an empty string (' ').
Stretch	Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
Strikethrough	Integer	True if Strikethrough style is enabled.
Strings	String	Character format tag of equation font to apply to Math Strings
StructuredApplication	String	String the identifies the name of the SGML or Xml application associated with the document. FM 7.0 or greater
StructuredApplicationForOpen	String	FM 10 or greater
StyleOverrides (Read-Only)	Integer	Style condition indicators for conditional text: CnDoubleUnderline CnNoOverride CnOverline CnSingleUnderline CnStrikeThrough
SubScriptShift	Metric	Baseline offset of subscripts expressed as percentage of current font size (metric 1% to 1000%)
SubScriptSize	Metric	Scaling factor for subscripts expressed as percentage of current font size (metric 1% to 1000%)
SubScriptStretch	Metric	Character stretch (set width) for subscripts expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
SuperScriptShift	Metric	Baseline offset of superscripts expressed as percentage of current font size (metric 1% to 1000%)
SuperScriptSize	Metric	Scaling factor for superscripts expressed as percentage of the current font size (metric 1% to 1000%)
SuperScriptStretch	Metric	Character stretch (set width) for superscripts expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
Symbols	String	Character format tag of equation font to apply to Math Symbols

Table 175: Document Properties (Page 20 of 24)

Property Name	Data Type	Description
SymbolsList (Read-Only)	StringList	List of math symbol fonts used in Equation Fonts dialog box
TblFnCellPosition	Integer	Placement of footnote number in footnote text: FnPosSuper: Superscript FnPosBaseline: Baseline FnPosSub: Subscript
TblFnCellPrefix	String	Prefix to appear before table footnote number in table cell
TblFnCellSuffix	String	Suffix to appear after table footnote number in table cell
TblFnCustNumString	String	Characters for custom document footnote numbers
TblFnFmt	String	Paragraph tag of table footnote
TblFnNumStyle	Integer	Footnote numbering style for tables in document: FnNumNumeric: Arabic FnNumRomanUC: Roman uppercase FnNumRomanLC: Roman lowercase FnNumAlphaUC: Alphabetic uppercase FnNumAlphaLC: Alphabetic lowercase FnNumKanji: Kanji characters FnNumZenkaku: Zenkaku FnNumZenkakuUC: Zenkaku uppercase FnNumZenkakuLC: Zenkaku lowercase FnNumKanjiKazu: Kazu FnNumDaiji: Daiji FnNumCustom: Custom numbering
TblFnPosition	Integer	Placement of footnote number in text: FnPosSuper: Superscript FnPosBaseline: Baseline FnPosSub: Subscript
TblFnPrefix	String	Prefix to appear before number in table footnote
TblFnSuffix	String	Suffix to appear after number in table footnote
TextSelection	TextRange	The currently selected text range or insertion point in the document. For information on getting and setting the text selection or insertion point in a document.
TopMargin	Metric	Top page margin
TrackChangesAddedColor	Object	The color object(Color)
TrackChangesDeletedColor	Object	The color object(Color)
TrackChangesOn	Integer	True if track changes is enabled. FM 8.0 or greater
TrapwiseCompatibility	Integer	True if Trapwise Compatibility is enabled. Setting this to True automatically sets PrintToFile to True and PrintSep to False.

Table 175: Document Properties (Page 21 of 24)

Property Name	Data Type	Description
Underlining	Integer	Type of underlining: CbNoUnderline CbSingleUnderline CbDoubleUnderline CbNumericUnderline
Untouchable	Integer	False by default. Setting this to True allows your client to modify a document without setting <code>DocIsModified</code> to True.
UseBkColor	Integer	True/False. FM 10 or greater
UseInitialStructure	Integer	True if Structured FrameMaker inserts initial structure for new elements.
UseInitialStructureOfAutoInsertedElements	Integer	True/False. FM 10 or greater
UseSepOverride	Integer	True if <code>SepOverride</code> overrides default.
Variables	String	Character format tag of equation font to apply to Math Variables
ViewBorders	Integer	True if Borders is enabled.
ViewDisplayUnits	Metric	The Metric equivalent of one unit in the current Display Units. For example, if Display Units is points, this returns 65536.
ViewFontSizeUnits	Metric	The Metric equivalent of one unit in the current Font Size Unit. Font size units can be either Points or Q. If Points, this returns 65536. If Q, this returns 47098
ViewGrid	Integer	True if View Grid is enabled.
ViewGridUnits	Metric	Units for Grid Lines.
ViewNoGraphics	Integer	True if Graphics is not enabled.
ViewOnlyDeadCodes	UIntList	F-codes that can't be executed in the document
ViewOnlyMenuBar	Object	The document's menu bar object when the document is locked (Menu)
ViewOnlyMenuBar	Object	If the document has a specific menu bar, the menubar's menu object for the document; otherwise 0
ViewOnlySelect	Integer	Specifies whether user can select text or graphics in the document: VosUserOnly: the user can select text when pressing modifier keys, and link targets (cross-reference sources and newlinks) do not highlight VosNone: the user can't select text, and links targets do not highlight VosYes: the user can select text (using modifier keys) and link targets are highlighted
ViewOnlyWinBorders	Integer	True if the document has normal document borders; False if the document scroll bars and border buttons are suppressed
ViewOnlyWinMenuBar	Integer	True if the document has a document window menu bar
ViewOnlyWinPalette	Int	True if document acts like a palette when it is View Only.
ViewOnlyWinPalette	Integer	True if the document is a palette
ViewOnlyWinPopup	Integer	True if the document window pop-up menu is available.

Table 175: Document Properties (Page 22 of 24)

Property Name	Data Type	Description
ViewOnlyXRef	Integer	Specifies the behavior of cross-references in the document: VoxNotActive : cross-references are not active VoxGotoBehavior : internal cross-references are active VoxOpenBehavior : external cross-references are active VoxAlert : alert appears when cross-reference is clicked
ViewPageScrolling	Integer	Page scrolling: ScrollVariable ScrollHorizontal ScrollVertical ScrollFacing
ViewRulers	Integer	True if Rulers is enabled.
ViewRulerUnits	Metric	Units for rulers display.
ViewTextSymbols	Integer	True if Text Symbols is enabled.
VolNumComputeMethod	Integer	Volume Numbering Computing Method: <i>FM 6.0 or greater</i> NumContinue : Continue the numbering from the previous volume. NumRestart : Use the number specified by VolumeNumber . NumSame : Use the same volume number as the previous file.
VolumeNumber	Integer	Volume Number: Use this as the volume number when the VolNumComputeMethod property is set to NumRestart <i>FM 6.0 or greater</i>
VolumeNumStyle	Integer	The numbering style of the volume: <i>FM 6.0 or greater</i> NumStyleNumeric : Arabic. NumStyleRomanUC : Roman, uppercase. NumStyleRomanLC : Roman, lowercase. NumStyleAlphaUC : Alphabetic, uppercase. NumStyleAlphaLC : Alphabetic, lowercase. NumStyleKanji : Kanji. NumStyleZenkaku : Zenkaku. NumStyleZenkakuUC : Zenkaku, uppercase. NumStyleZenkakuLC : Zenkaku, lowercase. NumStyleKanjiKazu : Kazu. NumStyleDaiji : Daiji. NumStyleText : Text NumStyleFarsiAlpha <i>FM 2017 or greater</i> NumStyleHebrewAlpha <i>FM 2017 or greater</i> NumStyleThaiAlpha <i>FM 2017 or greater</i> NumStyleThaiNumeric <i>FM 2017 or greater</i>
VolumeNumText	String	Volume Number text, if the VolumeNumStyle property is set to NumStyleText . <i>FM 6.0 or greater</i>
WideRubiSpaceForJapanese	Integer	Allowable values are: Wide Narrow Proportional
WideRubiSpaceForOther	Integer	Allowable values are: Wide Narrow Proportional

Table 175: Document Properties (Page 23 of 24)

Property Name	Data Type	Description
XmlDoc	Integer	True indicates that the current document corresponds to an XML document. (FM 7.0 or greater)
XmlDocType	String	The DOCTYPE parameter from the source XML. (FM 7.0 or greater)
XmlEncoding	String	The encoding parameter of the XML Declaration for the source XML document. The string is empty if no encoding is specified. If this property is set, the XML Declaration will contain the encoding parameter with this value when doing Save As XML. (FM 7.0 or greater)
XmlFileEncoding	String	The encoding that was detected for the source XML. If no encoding was specified for the source XML, XmlEncoding will be an empty string. In that case, if this string is set, it will determine the encoding to use when saving as XML. If XmlEncoding has a value, this string may be empty. (FM 7.0 or greater)
XmlLock	Integer	Undocumented. FM 11 or greater
XmlPublicId	String	The DOCTYPE public identifier for the source XML. (FM 7.0 or greater)
XmlStandAlone	Integer	An integer that specifies the XML standalone parameter for the XML document that was the source of this document. It can be one of the following values. XmlStandAloneYes XmlStandAloneNo XmlStandAloneNA This is declared in the XML Declaration. For a file with no XML declaration, the value is XmlStandAloneNoDec. For an XML declaration with no standalone parameter, the value is XmlStandAloneNone. (FM 7.0 or greater)
XmlStyleSheet	String	The XML stylesheet processing instruction to write out to XML when saving the book as XML. This value is not verified for the correct syntax. The string you set should not include the PI delimiters <? and ?>. For example, the string you supply for my.css would be: <code>'type="text\css" href="my.css"'</code> Only use this string to set a specific stylesheet specification. This value always returns an empty string. To get a list of stylesheet specifications associated with a book, use XmlStyleSheetList, below. (FM 7.0 or greater)
XmlStyleSheetList	StringList	A list of stylesheet processing instruction for the current book. One book can have more than one stylesheet specification associated with it. These values are not verified for the correct syntax. The strings should not include the PI delimiters <? and ?>. For example, the string you supply for my.css would be: <code>'type="text\css" href="my.css"'</code> Setting this property completely overwrites the current list. (FM 7.0 or greater)
XmlSystemId	String	The DOCTYPE system identifier for the source XML. (FM 7.0 or greater)

Table 175: Document Properties (Page 24 of 24)

Property Name	Data Type	Description
XmlUseBOM	Integer	An integer that specifies whether a byte order mark was detected when opening the source XML. It can be one of the following values. XmlUseBOMYes XmlUseBOMNo XmlUseBOMNA When saving as XML, this is set to XmlUseBOMYes. FrameMaker writes a byte order mark in the resulting XML. (FM 7.0 or greater)
XmlVersion	String	The XML version that was specified in the XML Declaration when the file was opened. If no XML version was specified, this will be an empty string. (FM 7.0 or greater)
XmlWellFormed	Integer	An integer that specifies whether the source XML qualified as well formed. It can be one of the following values. XmlWellFormedYes XmlWellFormedNo XmlWellFormedNA (FM 7.0 or greater)
XSLTProcessors	StringList	Undocumented. FM 11 or greater
XSLTTransformationScenarioFile	String	Undocumented. FM 11 or greater
Zoom	Metric	Zoom percentage of document (metric 25% to 1600%).

XMP Metadata

FrameMaker supports XMP Metadata, which is a protocol to store information about a file as encoded packets that are available to external applications. This information is similar to the information stored in the PDF Document Info dictionary. However, XMP data can contain fields that have no counterpart in PDF Document Info.

FrameMaker maps the values of string pairs in PDFDocInfo to XMP Metadata. If an external application modifies the document's metadata, these mapped fields will be updated in PDFDocInfo. Likewise, if you change a field in PDFDocInfo, then FrameMaker will update the encoded XMP packets to reflect this new information. The following table lists the supported fields. External indicates that the field value can be changed by your client, an external application, or the user interface. Internal indicates a field that FrameMaker maintains—you cannot modify its value.

Table 176: XMP Metadata

PDF Field:	XMP Field	Internal/External:
Author	dc:Creator	External
Title	dc:Title	External
Subject	dc:Description	External
Keywords	pdf:Keywords	External
Copyright	dc:Rights	External
Web Statement	xapRights:WebStatement	External

Table 176: XMP Metadata

PDF Field:	XMP Field	Internal/External:
Job Reference	xapBJ:JobRef	External
Marked	xapRights:Marked	External
Creation Date	xap:CreateDate	Internal
ModDate	xap:ModifyDate	Internal
Metadata Date	xap:MetadataDate	Internal
Creator	pdf:CreatorTool	Internal (FrameMaker—not displayed in dialog box)

You can modify XMP data directly for a document by setting a value to the FileInfoPacket document property. XMP uses the RDF syntax—see <http://www.w3.org/1999/02/22-rdf-syntax-ns#>, or print the FileInfoPacket to the console to see the XMP syntax.

HypertextParseErr Table

The following table shows error codes that can be set to HypertextParseErr.

Table 177: Hypertext Parse Error codes

Value	Description
HypertextBadSyntaxPathSpec	File reference expected for this command, but no valid file path found
HypertextEmptyCommand	Hypertext string is empty
HypertextExpectedANumberParam	Command expected a number but got text; check HypertextParseBadParam
HypertextExtraArguments	More than the required number of arguments for the command; extra arguments were ignored
HypertextHelpDirNotFound	Default help directory either does not exist (help was not installed) or cannot be found
HypertextMissingArguments	One or more arguments required for the command is missing
HypertextSyntaxOk	No parse errors
HypertextUnanchoredPartialPath	File reference is relative to the current document, but the current document has not been saved; file location could not be calculated
HypertextUnrecognizedCommand	Cannot map the first keyword to an existing HypertextParsedCmdCode

HypertextValidateErr Table

The following table shows error codes that can be set to HypertextValidateErr.

Table 178: Hypertext validation error codes (Page 1 of 2)

Value	Description
HypertextBadMatrixSize	One or both of the matrix dimensions is bad; must be between 1 and 99
HypertextCantOpenDestFile	Can't open the file; perhaps you don't have permission, or the file is locked

Table 178: Hypertext validation error codes (Page 2 of 2)

Value	Description
<code>HypertextCommandIllegalWithinPopup</code>	Invalid command in the popup command's reference flow; for example, matrix or newlink
<code>HypertextDestinationLinkNotFound</code>	The referenced file is valid, but can't find the named link within it
<code>HypertextFcodeInvalid</code>	Invalid FCode in the hypertext command
<code>HypertextFileNotMakerDoc</code>	The referenced file is not made by FrameMaker
<code>HypertextFileNotRegular</code>	The referenced file could not be found, or is not a regular file; for example, it could be a directory name
<code>HypertextFlowMissingLines</code>	The reference flow for a matrix or popup command is missing one or more lines
<code>HypertextMatrixCommandInvalid</code>	One of the commands in the reference page flow for a matrix command has a parse or validation error
<code>HypertextMissingPopupLabelItem</code>	One entry in the popup command's reference flow has no text in it
<code>HypertextEmptyLineInMiddleOfPopup</code>	
<code>HypertextMissingPopupMarker</code>	At least one entry in the popup command's reference flow has no hypertext marker in it
<code>HypertextNoNamedFlow</code>	Can't find the named reference flow for a matrix or popup command
<code>HypertextObjectIDNotFound</code>	A link to an object, but can't find the object
<code>HypertextPageNameNotFound</code>	The referenced file is valid, but can't find the specified page
<code>HypertextRecursiveFlow</code>	The reference flow for a matrix or popup command contains nested popup or matrix commands that name a parent reference flow
<code>HypertextUnrecognizedObjectType</code>	The referenced file is valid, but the link is to an object with an unrecognized object type
<code>HypertextUsesDefaultText</code>	Default text was found as an argument; are you sure the default text is what you want?
<code>HypertextValid</code>	No validation errors

Hypertext Command Code Table

The following table shows the possible values for `HypertextParsedCmdCode`

Table 179: Hypertext Parsed Command Codes

Value	Description
CmdAlert	alert command
CmdAlertTitle	alerttitle command
CmdError	Parser is in an error state
CmdExit	exit command
CmdGoToLink	gotolink command
CmdGoToLinkFitWin	gotolinkfitwin command
CmdGoToNew	gotonew command
CmdGoToObjectId	gotoObjectId command
CmdGoToObjectIdFitWin	gotoObjectIdfitwin command
CmdGoToPage	gotopage command
CmdMatrix	matrix command
CmdMessage	message command
CmdNewLink	newlink command
CmdNextPage	nextpage command
CmdNoop	Command causes no event
CmdOpenLink	openlink command
CmdOpenLinkFitWin	openlinkfitwin command
CmdOpenNew	opennew command
CmdOpenObjectId	openObjectId command
CmdOpenObjectIdFitWin	openObjectIdfitwin command
CmdOpenPage	openpage command
CmdPopup	popup command
CmdPreviousLink	previouslink command
CmdPreviousLinkFitWin	previouslinkfitwin command
CmdPreviousPage	previouspage command
CmdQuit	quit command
CmdQuitAll	quitall command
CmdUnknown	Unknown command

The following table shows the possible values for HypertextParsedCmdDest

Table 180: Hypertext Parsed Command Destination values

Value	Description
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
DestMarkerNewLink	Destination is a newlink
DestNowhere	No destination found
DestObjectId	Destination is an object (usually for generated hypertext commands)
DestPageNum	Destination is a named page (usually a page number)
DestXRef	Destination is a cross-reference

Link Destination Object Type Table

The following table shows the possible values for HypertextParsedCmdDestObjType.

Table 181: Hypertext Parsed Command Destination Object Types

Value	Description
ObjectDataLink	Object is subscribed data
ObjectGraphic	Object is a graphic
ObjectMarker	Object is a marker
ObjectPgf	Object is a paragraph
ObjectTextInset	Object is a text inset
ObjectUnknown	Unknown or invalid object
ObjectXref	Object is a cross-reference

Flows

The Flow object is used to represent each flow in the document.

Object name - Flow

Flow objects have the following properties.

Table 182: Flow Properties

Property Name	Data Type	Description
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection <i>(Read-Only)</i>	Integer	FM 2015 or greater DirLtr DirRtl.
FlowIsAutoConnect	Integer	True if Autoconnect is enabled.
FlowIsFeathered	Integer	True if Feather is enabled.
FlowIsPostScript	Integer	True if flow is PostScript code.
FlowIsSynchronized	Integer	True if Baseline Synchronization is enabled.
FirstPgInFlow <i>(Read-Only)</i>	Object	First paragraph in the first text frame (FirstTextFrameInFlow) in flow (TextFrame Object).
FirstTextFrameInFlow <i>(Read-Only)</i>	Object	First text frame in flow (TextFrame Object).
HighestLevelElement	Object	Highest level element in the flow (Flow Object).
LastPgInFlow <i>(Read-Only)</i>	Object	First paragraph in the first text frame (FirstTextFrameInFlow) in flow (TextFrame Object).
LastTextFrameInFlow <i>(Read-Only)</i>	Object	Last text frame in flow (TextFrame Object).
MaxInterlinePadding	Metric	Maximum interline spacing.
MaxInterPgfPadding	Metric	Maximum interparagraph spacing.
MinHang	Metric	Maximum character height for synchronization of first line in column. If characters exceed this height, FrameMaker doesn't synchronize the first line.
Name	String	Name of flow tag.
NextFlowInDoc <i>(Read-Only)</i>	Object	Next flow in document (Flow Object).
SideHeadRoomInFlow	Integer	True if Leave Room for Sideheads in Flow is enabled.
Spacing	Metric	Line spacing for synchronized baselines.
Text	String	The text for the entire flow object

Footnotes

The `Fn` object is used to represent a footnote.

Object name - `Fn`

`Fn` objects have the following properties.

Table 183: Footnote Properties

Property Name	Data Type	Description
<code>ContentHeight</code> (<i>Read-Only</i>)	Metric	The distance between the top of the footnote and the baseline of the last line in the footnote
<code>Element</code>	Object	If the footnote is in a FM+SGML document, the element containing the footnote (<code>Element</code> Object)
<code>FirstPgf</code> (<i>Read-Only</i>)	Object	First paragraph in the footnote (<code>Pgf</code> Object)
<code>FnAnchorString</code> (<i>Read-Only</i>)	String	The anchor string of the footnote. FM 9.0 or greater
<code>FnNum</code> (<i>Read-Only</i>)	Integer	Footnote number
<code>InTextFrame</code> (<i>Read-Only</i>)	Object	Text frame containing the footnote (<code>TextFrame</code> Object)
<code>InTextObj</code> (<i>Read-Only</i>)	Object	Text frame or cell used to format the footnote (<code>TextFrame</code> or <code>Cell</code> Object)
<code>LastPgf</code> (<i>Read-Only</i>)	Object	Last paragraph in the footnote (<code>Pgf</code> Object)
<code>NextFnInDoc</code> (<i>Read-Only</i>)	Object	Next footnote (<code>Fn</code> Object) in the document
<code>NextFn</code> (<i>Read-Only</i>)	Object	Next footnote in the text frame (<code>Fn</code> Object)
<code>Overflowed</code> (<i>Read-Only</i>)	Integer	True if the text in the footnote overflows
<code>PrevFn</code> (<i>Read-Only</i>)	Object	Previous footnote in the text frame (<code>Fn</code> Object)
<code>TextLoc</code> (<i>Read-Only</i>)	TextLoc	Text location of the footnote symbol

`Doc` objects also have properties that specify how all the footnotes in the document appear.

Graphic Objects

A variety of objects are known as Graphic properties. These are `AFrame` (anchored frame) `Arc`, `Ellipse`, `Group`, `Inset`, `Line`, `Math`, `MathML`, `Polygon`, `Polyline`, `Rectangle`, `RoundRect`, `TextFrame`, `Textline`, and `UnanchoredFrame`. The properties of these object are very similar.

Object name - AFrame

An `AFrame` object represents an anchored frame. `AFrame` objects have the following properties.

Table 184: Anchored Frame Properties (Page 1 of 3)

Property Name	Data Type	Description
<code>AFrameIsCropped</code>	<code>Integer</code>	True if Cropped is enabled
<code>AFrameIsFloating</code>	<code>Integer</code>	True if Floating is enabled
<code>Alignment</code>	<code>Integer</code>	Type of alignment: <code>AlignCenter</code> <code>AlignInside</code> <code>AlignOutside</code> <code>AlignLeft</code> <code>AlignRight</code>
<code>AnchorType</code>	<code>Integer</code>	Where frame is anchored: <code>AnchorBelow</code> <code>AnchorBottom</code> <code>AnchorInline</code> <code>AnchorRunIntoParagraph</code> <code>AnchorSubColFarthest</code> <code>AnchorSubColInside</code> <code>AnchorSubColLeft</code> <code>AnchorSubColNearest</code> <code>AnchorSubColOutside</code> <code>AnchorSubColRight</code> <code>AnchorTextframeFarthest</code> <code>AnchorTextframeInside</code> <code>AnchorTextframeLeft</code> <code>AnchorTextframeNearest</code> <code>AnchorTextframeOutside</code> <code>AnchorTextframeRight</code> <code>AnchorTop</code>
<code>Angle</code>	<code>Metric</code>	Angle of the object's rotation.
<code>ArrowBaseAngle</code>	<code>Integer</code>	Arrowhead base angle in degrees.
<code>ArrowLength</code>	<code>Metric</code>	Arrowhead length (always rounded down to the nearest 1/256 point).
<code>ArrowScaleFactor</code>	<code>Metric</code>	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
<code>ArrowScaleHead</code>	<code>Integer</code>	True if arrowhead is scaled as the line width changes.
<code>ArrowTipAngle</code>	<code>Integer</code>	Arrowhead tip angle in degrees.

Table 184: Anchored Frame Properties (Page 2 of 3)

Property Name	Data Type	Description
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BaselineOffset	Metric	Baseline offset
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FirstGraphicInFrame (Read-Only)	Object	First object in frame
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
InTextFrame (Read-Only)	Object	Text frame in which anchored frame appears (TextFrame Object)
InTextObj (Read-Only)	Object	Column or text frame in which anchored frame appears (SubCol or TextFrame Object)
IsHotspot	Integer	Undocumented. FM 11 or greater
LastGraphicInFrame (Read-Only)	Object	Last object in frame
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	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. You can't set LocX for anchored frames.
LocY	Metric	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. You can't set LocY for anchored frames.
MifFileEntity	String	Undocumented. FM 11 or greater

Table 184: Anchored Frame Properties (Page 3 of 3)

Property Name	Data Type	Description
MifFileName	String	Undocumented. FM 11 or greater
MifFileOrigFormat	String	Undocumented. FM 11 or greater
NextAFrame (Read-Only)	Object	Next anchored frame in text frame (AFrame Object)
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevAFrame (Read-Only)	Object	Previous anchored frame in text frame (AFrame Object)
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
SideOffset	Metric	Near side offset
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TextLoc (Read-Only)	TextLoc	Text location of the anchor symbol
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

For different anchored frame types (AnchorType), certain properties of the AFrame object must have specific values. For example, if the AnchorType is AnchorRunIntoParagraph, the SideOffset must be 0. The following table lists the constraints on properties for different anchored frame types.

Table 185: Anchored Frame Property Constraints (Page 1 of 2)

AnchorType	Property constraints
AnchorInline	SideOffset = 0
AnchorTop	SideOffset = 0
AnchorBelow	BaselineOffset = 0
AnchorBottom	

Table 185: Anchored Frame Property Constraints (Page 2 of 2)

AnchorType	Property constraints
AnchorRunIntoParagraph	SideOffset = 0 BaselineOffset = 0 AFrameIsFloating = 0 AFrameIsCropped = 0
AnchorSubcolFarthest AnchorSubcolInside AnchorSubcolLeft AnchorSubcolNearest AnchorSubcolOutside AnchorSubcolRight AnchorTextframeFarthest AnchorTextframeInside AnchorTextframeLeft AnchorTextframeNearest AnchorTextframeOutside AnchorTextframeRight	AFrameIsFloating = 0 AFrameIsCropped = 0

Object name - Arc

An Arc object represents an arc. Arc objects have the following properties.

Table 186: Arc Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if ArrowScaleHead is False.
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
DTheta	Metric	Arc angle length (-360 degree to 360 degree)

Table 186: Arc Properties (Page 2 of 3)

Property Name	Data Type	Description
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object).
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.

Table 186: Arc Properties (Page 3 of 3)

Property Name	Data Type	Description
PrevGraphicInGroup (Read-Only)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
Theta	Metric	Start angle (0 degree to 360 degree)
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Ellipse

An `Ellipse` object represents an ellipse. `Ellipse` objects have the following properties.

Table 187: Ellipse Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	<code>True</code> if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear

Table 187: Ellipse Properties (Page 2 of 3)

Property Name	Data Type	Description
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches)
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	Object	Previous graphic object in the group.
RectangleIsSmoothed (Read-Only)	Integer	True if smoothing is enabled. This property is always True for Ellipse objects.

Table 187: Ellipse Properties (Page 3 of 3)

Property Name	Data Type	Description
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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - GraphicsFmt (FM 11 or greater)

GraphicsFmt objects have the following properties. Several of these properties are fixed and are always present like most objects, however, most of these properties come in pairs. One member of the pair has the prefix 'Use'. The properties with 'Use' as the prefix are always present. However, the corresponding property is only present when the 'Use' property is set to True. For example, the Alignment property is not present, unless the UseAlignment property is set to True.

Table 188: GraphicsFmt Properties (Page 1 of 6)

Property Name	Data Type	Description
AFrameIsCropped	Integer	True if Cropped is enabled. (This Property is only present when the corresponding 'Use' property is set to True.)
AFrameIsFloating	Integer	True if Floating is enabled. (This Property is only present when the corresponding 'Use' property is set to True.)
Alignment	Integer	Type of alignment: AlignCenter AlignInside AlignOutside AlignLeft AlignRight (This Property is only present when the corresponding 'Use' property is set to True.)

Table 188: GraphicsFmt Properties (Page 2 of 6)

Property Name	Data Type	Description
AnchorType	Integer	Where frame is anchored: AnchorBelow AnchorBottom AnchorInline AnchorRunIntoParagraph AnchorSubColFarthest AnchorSubColInside AnchorSubColLeft AnchorSubColNearest AnchorSubColOutside AnchorSubColRight AnchorTextframeFarthest AnchorTextframeInside AnchorTextframeLeft AnchorTextframeNearest AnchorTextframeOutside AnchorTextframeRight AnchorTop (This Property is only present when the corresponding 'Use' property is set to True.)
Angle	Metric	Angle of the object's rotation. (This Property is only present when the corresponding 'Use' property is set to True.)
BaselineOffset	Metric	Baseline offset (This Property is only present when the corresponding 'Use' property is set to True.)
BorderWidth	Metric	Border width (0.015 pt to 360 pt). (This Property is only present when the corresponding 'Use' property is set to True.)
ColGapWidth	Metric	Gap between columns (0 to 50 inches). (This Property is only present when the corresponding 'Use' property is set to True.)
Color	Object	The spot color (Color Object). (This Property is only present when the corresponding 'Use' property is set to True.)
ColumnsAreBalanced	Integer	True if terminal and underfilled columns in the flow are balanced. (This Property is only present when the corresponding 'Use' property is set to True.)
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
DTheta	Metric	Arc angle length (-360 degree to 360 degree) (This Property is only present when the corresponding 'Use' property is set to True.)

Table 188: GraphicsFmt Properties (Page 3 of 6)

Property Name	Data Type	Description
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear (This Property is only present when the corresponding 'Use' property is set to True.)
FlowIsAutoConnect	Integer	True if Autoconnect is enabled. (This Property is only present when the corresponding 'Use' property is set to True.)
FlowIsPostScript	Integer	True if flow is PostScript code. (This Property is only present when the corresponding 'Use' property is set to True.)
Height	Metric	Height of object (0.125 pt to 3600 pt). (This Property is only present when the corresponding 'Use' property is set to True.)
InsetDpi	Integer	Scaling information for bitmap file (corresponds to the value specified in the Image File Scaling Options dialog box when the graphics file is imported). (This Property is only present when the corresponding 'Use' property is set to True.)
LocX	Metric	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. You can't set LocX for anchored frames. (This Property is only present when the corresponding 'Use' property is set to True.)
LocY	Metric	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. You can't set LocY for anchored frames. (This Property is only present when the corresponding 'Use' property is set to True.)
MathSize	Integer	Equation size: MathLarge MathMedium MathSmall (This Property is only present when the corresponding 'Use' property is set to True.)
NextGraphicsFmtInDoc (Read-Only)	Object	Next graphic format in the document.
NumColumns	Integer	The number of columns in the underlying column grid (1–10). (This Property is only present when the corresponding 'Use' property is set to True.)
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor (This Property is only present when the corresponding 'Use' property is set to True.)

Table 188: GraphicsFmt Properties (Page 4 of 6)

Property Name	Data Type	Description
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear (This Property is only present when the corresponding 'Use' property is set to True.)
Radius	Metric	Radius of corner; 0 for a square corner. (This Property is only present when the corresponding 'Use' property is set to True.)
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 TrNone TrContour TrBBox (This Property is only present when the corresponding 'Use' property is set to True.)
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap. (This Property is only present when the corresponding 'Use' property is set to True.)
SideHeadGap	Metric	Gap between side head area and body text area (0 to 50 inches). (This Property is only present when the corresponding 'Use' property is set to True.)
SideHeadPlacement	Integer	Placement of side heads relative to columns in the text frame: ShLeft ShRight ShInside ShOutside (This Property is only present when the corresponding 'Use' property is set to True.)
SideHeadWidth	Metric	Width of side head area for the text frame (0 to 50 inches). (This Property is only present when the corresponding 'Use' property is set to True.)
SideOffset	Metric	Near side offset (This Property is only present when the corresponding 'Use' property is set to True.)
StyleTag	String	Name of the Graphics Format
TextLineType	Integer	Type of text line: TextlineLeft TextlineRight TextlineCenter TextlineMath (This Property is only present when the corresponding 'Use' property is set to True.)

Table 188: GraphicsFmt Properties (Page 5 of 6)

Property Name	Data Type	Description
Theta	Metric	Start angle (0 degree to 360 degree) (This Property is only present when the corresponding 'Use' property is set to True.)
TintPercent	Metric	The tint percentage (This Property is only present when the corresponding 'Use' property is set to True.)
UseAFrameIsCropped	Integer	
UseAFrameIsFloating	Integer	
UseAlignment	Integer	
UseAnchorType	Integer	
UseAngle	Integer	
UseBaselineOffset	Integer	
UseBorderWidth	Integer	
UseColGapWidth	Integer	
UseColor	Integer	
UseColumnsAreBalanced	Integer	
UseDTheta	Integer	
UseFill	Integer	
UseFlowIsAutoConnect	Integer	
UseFlowIsPostScript	Integer	
UseHeight	Integer	
UseInsetDpi	Integer	
UseLocX	Integer	
UseLocY	Integer	
UseMathMLApplyPgfStyle	Integer	FM 2015 or greater
UseMathMLComposeDpi	Integer	FM 2015 or greater
UseMathMLDpi	Integer	FM 2015 or greater
UseMathMLFontSize	Integer	FM 2015 or greater
UseMathMLInline	Integer	FM 2015 or greater
UseMathSize	Integer	
UseNumColumns	Integer	
UseOverprint	Integer	
UsePen	Integer	
UseRadius	Integer	
UseRunaround	Integer	

Table 188: GraphicsFmt Properties (Page 6 of 6)

Property Name	Data Type	Description
UseRunaroundGap	Integer	
UseSideHeadGap	Integer	
UseSideHeadPlacement	Integer	
UseSideHeadWidth	Integer	
UseSideOffset	Integer	
UseTheta	Integer	
UseTextLineType	Integer	
UseTintPercent	Integer	
UseWidth	Integer	
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Group

An `Group` object represents a set of grouped objects. `Group` objects have the following properties.

Table 189: Group Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	<code>True</code> if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear

Table 189: Group Properties (Page 2 of 3)

Property Name	Data Type	Description
FirstGraphicInGroup (Read-Only)	Object	First object in the group
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
LastGraphicInGroup (Read-Only)	Object	Last object in the group
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	Object	Previous graphic object in the group.

Table 189: Group Properties (Page 3 of 3)

Property Name	Data Type	Description
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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Inset

An `Inset` object represents an imported graphic. `Inset` objects have the following properties.

Table 190: Inset Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (<code>AFrame</code> or <code>UnanchoredFrame</code> Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.

Table 190: Inset Properties (Page 2 of 3)

Property Name	Data Type	Description
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
InsetCuePointList	IntList	Undocumented. FM 11 or greater
InsetDpi	Integer	Scaling information for bitmap file (corresponds to the value specified in the Image File Scaling Options dialog box when the graphics file is imported).
InsetEditor	String	Name of application to call to edit inset or imported object.
InsetFile	String	Platform-specific pathname if the inset is an external inset, or a null string (" ") if it is internal. The pathname can be document-relative.
InsetFileOrigName	String	
InsetGenericData	UByteList	FM 10 or greater
InsetGfxActiveInPdf	String	Undocumented. FM 11 or greater
InsetGfxName	String	Undocumented. FM 11 or greater
InsetGfxPlayWindowInPdf	Integer	Undocumented. FM 11 or greater
InsetInfo	UByteList	
InsetIsFixedSize	Integer	True if scaling of bitmap file is inhibited.
InsetIsFlippedSideways	Integer	True if inset is flipped about the vertical axis.
InsetJavaScriptAttached	Integer	Undocumented. FM 11 or greater
InsetJavaScriptFile	Integer	Undocumented. FM 11 or greater
InsetLinkToText	String	Undocumented. FM 11 or greater
InsetMonikerFilePath	String	Undocumented. FM 11 or greater
InsetMonikerPath	String	Undocumented. FM 11 or greater
InsetPosterFile	String	FM 10 or greater
InsetSaveFacetToFile	String	Undocumented. FM 11 or greater
InsetU3dAnimationList	StringList	Undocumented. FM 11 or greater
InsetU3dPartList	StringList	Undocumented. FM 11 or greater
InsetU3dViewList	StringList	Undocumented. FM 11 or greater
InsetUpdater	String	Not currently implemented.
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare

Table 190: Inset Properties (Page 3 of 3)

Property Name	Data Type	Description
LocX	Metric	Distance from the left side of the parent frame (–216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (–216 inches to 216 inches).
PrevGraphicInFrame	Object	Previous graphic object in the frame.
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalDpi	Integer	Original height. FM 2017 or greater
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInGroup (Read-Only)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Line

An `Line` object represents a line. `Line` objects have the following properties.

Table 191: Line Properties (Page 1 of 2)

Property Name	Data Type	Description
<code>Angle</code>	<code>Metric</code>	Angle of the object's rotation.
<code>ArrowBaseAngle</code>	<code>Integer</code>	Arrowhead base angle in degrees.
<code>ArrowLength</code>	<code>Metric</code>	Arrowhead length (always rounded down to the nearest 1/256 point).
<code>ArrowScaleFactor</code>	<code>Metric</code>	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
<code>ArrowScaleHead</code>	<code>Integer</code>	<code>True</code> if arrowhead is scaled as the line width changes.
<code>ArrowTipAngle</code>	<code>Integer</code>	Arrowhead tip angle in degrees.
<code>ArrowType</code>	<code>Integer</code>	Arrowhead style: <code>ArrowStick</code> <code>ArrowHollow</code> <code>ArrowFilled</code>
<code>BorderWidth</code>	<code>Metric</code>	Border width (0.015 pt to 360 pt).
<code>Color</code>	<code>Object</code>	The spot color (<code>Color</code> Object).
<code>Dash</code>	<code>MetricList</code>	Dash style (see the description below).
<code>Fill</code>	<code>Integer</code>	The fill pattern (numbers between 0 and 15). The fill patterns: <code>FillBlack</code> <code>FillWhite</code> <code>FillClear</code>
<code>FrameParent</code>	<code>Object</code>	Frame containing the graphic object (<code>AFrame</code> or <code>UnanchoredFrame</code> Object).
<code>GraphicCantBeSelected</code>	<code>Integer</code>	<code>True</code> if the graphic object can't be selected.
<code>GraphicIsSelected</code>	<code>Integer</code>	<code>True</code> if the graphic object is selected.
<code>GroupParent</code>	<code>Object</code>	Group that the object is in (<code>Group</code> Object). Anchored and unanchored frames do not have this property.
<code>HeadArrow</code>	<code>Integer</code>	Arrowhead at end of line (<code>True</code> if line has arrowhead).
<code>Height</code>	<code>Metric</code>	Height of object (0.125 pt to 3600 pt).
<code>HotspotCmdStr</code>	<code>String</code>	Undocumented. FM 11 or greater
<code>HotspotTitle</code>	<code>String</code>	Undocumented. FM 11 or greater
<code>IsHotspot</code>	<code>Integer</code>	Undocumented. FM 11 or greater
<code>LineCap</code>	<code>Integer</code>	Type of line end: <code>CapButt</code> <code>CapRound</code> <code>CapSquare</code>
<code>LocX</code>	<code>Metric</code>	Distance from the left side of the parent frame (-216 inches to 216 inches).
<code>LocY</code>	<code>Metric</code>	Distance from the top of the parent frame (-216 inches to 216 inches).

Table 191: Line Properties (Page 2 of 2)

Property Name	Data Type	Description
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
NumPoints (Read-Only)	Integer	Number of vertices. The default is 2 (the line's start point and end point).
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
Points	PointList	Array of x-y coordinate pairs that specify the line's vertices. The default coordinate pairs are for the line's start point and end point.
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Math

An `Math` object represents an equation.

`Doc` objects have properties that specify how all the equations in a document appear.

Table 192: Math Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if ArrowScaleHead is False .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BasePointX	Metric	Horizontal placement of text line base point relative to left side of frame
BasePointY	Metric	Vertical placement of text line base point relative to top of frame
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).

Table 192: Math Properties (Page 2 of 3)

Property Name	Data Type	Description
LocY	Metric	Distance from the top of the parent frame (–216 inches to 216 inches).
MathFullForm	String	String representing the expression
MathMLExportEntitiesAsValues (Read-Only)	Integer	True or False:
MathSize	Integer	Equation size: MathLarge MathMedium MathSmall
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).

Table 192: Math Properties (Page 3 of 3)

Property Name	Data Type	Description
TextLineType	Integer	Type of text line: TextlineLeft TextlineRight TextlineCenter TextlineMath
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - MathML (FM 12 or greater)

An MathML object represents a MathML equation.

Table 193: MathML Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if ArrowScaleHead is False .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.

Table 193: MathML Properties (Page 2 of 3)

Property Name	Data Type	Description
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented.
HotspotTitle	String	Undocumented.
IsHotspot	Integer	Undocumented.
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
MathMLComposeDpi	Integer	Compose DPI.
MathMLData	StringList	Undocumented
MathMLDpi	Integer	DPI.
MathMLFile	String	Undocumented
MathMLFontSize	Integer	Font size.
MathMLIsFixedSize	Integer	True if fixed size, false otherwise.
MathMLIsFlippedSideways	Integer	True if flipped sideways, false otherwise.
MathMLIsInverted	Integer	True if inverted, false otherwise.
MathMLXmlData	String	Xml data for the equation.
NextGraphicInDoc <i>(Read-Only)</i>	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup <i>(Read-Only)</i>	Object	Next graphic object in the group.
NextSelectedGraphicInDoc <i>(Read-Only)</i>	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear

Table 193: MathML Properties (Page 3 of 3)

Property Name	Data Type	Description
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (<i>Read-Only</i>)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TextLineType	Integer	Type of text line: TextlineLeft TextlineRight TextlineCenter TextlineMath
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Polygon

An `Polygon` object represents a polygon. `Polygon` objects have the following properties.

Table 194: Polygon Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	<code>True</code> if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).

Table 194: Polygon Properties (Page 2 of 3)

Property Name	Data Type	Description
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
NumPoints	Integer	Number of the polygon's vertices
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor

Table 194: Polygon Properties (Page 3 of 3)

Property Name	Data Type	Description
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
Points (<i>Read-Only</i>)	PointList	Array of x-y coordinate pairs that specify the polygon's vertices
PolyIsBezier	Integer	True if polygon is smoothed
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (<i>Read-Only</i>)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Polyline

An `Polyline` object represents a polyline. `Polyline` objects have the following properties.

Table 195: Polyline Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled

Table 195: Polyline Properties (Page 2 of 3)

Property Name	Data Type	Description
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
NumPoints	Integer	Number of the polyline's vertices
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor

Table 195: Polyline Properties (Page 3 of 3)

Property Name	Data Type	Description
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
Points (<i>Read-Only</i>)	PointList	Array of x-y coordinate pairs that specify the polyline's vertices
PolyIsBezier	Integer	True if polyline is a Bezier curve
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (<i>Read-Only</i>)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - Rectangle

An `Rectangle` object represents a rectangle. `Rectangle` objects have the following properties.

Table 196: Rectangle Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled

Table 196: Rectangle Properties (Page 2 of 3)

Property Name	Data Type	Description
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor

Table 196: Rectangle Properties (Page 3 of 3)

Property Name	Data Type	Description
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	Object	Previous graphic object in the group.
RectangleIsSmoothed	Integer	True if smoothing is enabled
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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - RoundRect

An `RoundRect` object represents a rounded rectangle. `RoundRect` objects have the following properties.

Table 197: Round Rectangle Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).

Table 197: Round Rectangle Properties (Page 2 of 3)

Property Name	Data Type	Description
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear

Table 197: Round Rectangle Properties (Page 3 of 3)

Property Name	Data Type	Description
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	Object	Previous graphic object in the group.
Radius	Metric	Radius of corner; 0 for a square corner
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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - TextFrame

An `TextFrame` object represents a text frame. `TextFrame` objects have the following properties.

Table 198: Text Frame Properties (Page 1 of 4)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	<code>True</code> if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
ColGapWidth	Metric	Gap between columns (0 to 50 inches).
Color	Object	The spot color (<code>Color</code> Object).
ColumnsAreBalanced	Integer	<code>True</code> if terminal and underfilled columns in the flow are balanced.
Dash	MetricList	Dash style (see the description below).

Table 198: Text Frame Properties (Page 2 of 4)

Property Name	Data Type	Description
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FirstAFrame (Read-Only)	Object	First anchored frame in the text frame (AFrame Object).
FirstCell (Read-Only)	Object	First table cell in the text frame (Cell Object).
FirstFn (Read-Only)	Object	First footnote in the text frame (Fn Object).
FirstPgf (Read-Only)	Object	First paragraph in the text frame (Pgf Object).
FirstSubCol (Read-Only)	Object	First column in the text frame (SubCol Object).
Flow (Read-Only)	Object	Flow containing the text frame (Flow Object).
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsButton	Integer	True if the text frame is a hypertext button.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LastAFrame (Read-Only)	Object	Last anchored frame in the text frame (AFrame Object).
LastCell (Read-Only)	Object	Last table cell in the text frame (Cell Object).
LastFn (Read-Only)	Object	Last footnote in the text frame (Fn Object).
LastPgf (Read-Only)	Object	Last paragraph in the text frame (Pgf Object).
LastSubCol (Read-Only)	Object	Last column in the text frame (SubCol Object).

Table 198: Text Frame Properties (Page 3 of 4)

Property Name	Data Type	Description
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (–216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (–216 inches to 216 inches).
NextGraphicInDoc (Read-Only)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup (Read-Only)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (Read-Only)	Object	Next selected graphic object in document.
NextTextFrameInFlow	Object	Next text frame in the flow (TextFrame Object).
NumColumns	Integer	The number of columns in the underlying column grid (1–10).
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original width. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	Object	Previous graphic object in the group.
PrevTextFrameInFlow	Object	Previous text frame in the flow (TextFrame Object).
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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
SideHeadGap	Metric	Gap between side head area and body text area (0 to 50 inches).

Table 198: Text Frame Properties (Page 4 of 4)

Property Name	Data Type	Description
SideHeadPlacement	Integer	Placement of side heads relative to columns in the text frame: ShLeft ShRight ShInside ShOutside
SideHeadWidth	Metric	Width of side head area for the text frame (0 to 50 inches).
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
Text	String	The text in a text frame.
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - TextLine

An `TextLine` object represents a text line. `TextLine` objects have the following properties.

Table 199: Textline Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if <code>ArrowScaleHead</code> is <code>False</code> .
ArrowScaleHead	Integer	<code>True</code> if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BasePointX	Metric	Horizontal placement of text line base point relative to left side of frame.
BasePointY	Metric	Vertical placement of text line base point relative to top of frame.
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (<code>Color</code> Object).
Dash	MetricList	Dash style (see the description below).

Table 199: Textline Properties (Page 2 of 3)

Property Name	Data Type	Description
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection <i>(Read-Only)</i>	Integer	FM 2015 or greater DirLtr DirRtl.
Fill	Integer	The fill pattern (numbers between 0 and 15. The fill patterns: FillBlack FillWhite FillClear
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
GroupParent	Object	Group that the object is in (Group Object). Anchored and unanchored frames do not have this property.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
NextGraphicInDoc <i>(Read-Only)</i>	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.
NextGraphicInGroup <i>(Read-Only)</i>	Object	Next graphic object in the group.
NextSelectedGraphicInDoc <i>(Read-Only)</i>	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater

Table 199: Textline Properties (Page 3 of 3)

Property Name	Data Type	Description
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (Read-Only)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
Text	String	The text in a text line.
TextLineType	Integer	Type of text line: TextlineLeft TextlineRight TextlineCenter TextlineMath
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

Object name - UnanchoredFrame

An `UnanchoredFrame` object represents an unanchored frame. `UnanchoredFrame` objects have the following properties.

Table 200: Unanchored Frame Properties (Page 1 of 3)

Property Name	Data Type	Description
Angle	Metric	Angle of the object's rotation.
ArrowBaseAngle	Integer	Arrowhead base angle in degrees.
ArrowLength	Metric	Arrowhead length (always rounded down to the nearest 1/256 point).

Table 200: Unanchored Frame Properties (Page 2 of 3)

Property Name	Data Type	Description
ArrowScaleFactor	Metric	Factor by which arrowhead is scaled as line width changes (always rounded down to nearest 1/16 point). It is not used if ArrowScaleHead is False .
ArrowScaleHead	Integer	True if arrowhead is scaled as the line width changes.
ArrowTipAngle	Integer	Arrowhead tip angle in degrees.
ArrowType	Integer	Arrowhead style: ArrowStick ArrowHollow ArrowFilled
BorderWidth	Metric	Border width (0.015 pt to 360 pt).
Color	Object	The spot color (Color Object).
Dash	MetricList	Dash style (see the description below).
Fill	Integer	The fill pattern (numbers between 0 and 15). The fill patterns: FillBlack FillWhite FillClear
FirstGraphicInFrame (<i>Read-Only</i>)	Object	First object in the frame (backmost object).
FrameParent	Object	Frame containing the graphic object (AFrame or UnanchoredFrame Object).
GraphicCantBeSelected	Integer	True if the graphic object can't be selected.
GraphicIsSelected	Integer	True if the graphic object is selected.
HeadArrow	Integer	Arrowhead at end of line (True if line has arrowhead).
Height	Metric	Height of object (0.125 pt to 3600 pt).
HotspotCmdStr	String	Undocumented. FM 11 or greater
HotspotTitle	String	Undocumented. FM 11 or greater
IsHotspot	Integer	Undocumented. FM 11 or greater
LastGraphicInFrame (<i>Read-Only</i>)	Object	Last object in the frame (frontmost object).
LineCap	Integer	Type of line end: CapButt CapRound CapSquare
LocX	Metric	Distance from the left side of the parent frame (-216 inches to 216 inches).
LocY	Metric	Distance from the top of the parent frame (-216 inches to 216 inches).
Name	String	If a reference frame, the frame's name.
NextGraphicInDoc (<i>Read-Only</i>)	Object	Next graphic object in the document.
NextGraphicInFrame	Object	Next graphic object in the frame.

Table 200: Unanchored Frame Properties (Page 3 of 3)

Property Name	Data Type	Description
NextGraphicInGroup (<i>Read-Only</i>)	Object	Next graphic object in the group.
NextSelectedGraphicInDoc (<i>Read-Only</i>)	Object	Next selected graphic object in document.
OriginalHeight	Metric	Original height. FM 2015 or greater
OriginalWidth	Metric	Original height. FM 2015 or greater
Overprint	Integer	Specifies the overprint settings for the object OverPrint KnockOut FromColor
PageFramePage (<i>Read-Only</i>)	Object	If the unanchored frame is a page frame, the page that it belongs to (HiddenPage, BodyPage, MasterPage, or RefPage object).
Pen	Integer	The pen pattern (numbers between 0 and 15). The constants for several pen patterns: FillBlack FillWhite FillClear
PrevGraphicInFrame	Object	Previous graphic object in the frame.
PrevGraphicInGroup (<i>Read-Only</i>)	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 TrNone TrContour TrBBox
RunaroundGap	Metric	If the object is a runaround object, the width of the runaround gap.
StyleTag	String	Undocumented. FM 11 or greater
TailArrow	Integer	Arrowhead at beginning of line (True if enabled).
TintPercent	Metric	The tint percentage
Width	Metric	Width of object (0.125 pt to 3600 pt).

IMPORTANT: The `Dash` property specifies a dash pattern that is repeated for the length of an object's border. The pattern is stored in an `MetricList` structure. The 1st member of the `MetricList` stores the length of the first dash; the 2nd member stores the following space; the 3rd member stores the next dash; and so on for an even number of member.

Inline Components (FM 2015 or greater)

An `InlinComponent` object represents a special component.

Object name - InlineComponent

`InlineComponent` objects have the following properties.

Table 201: InlineComponent Properties

Property Name	Data Type	Description
<code>ExtractTags</code>	<code>StringList</code>	The list paragraph tags used for the minitoc.
<code>InlineComponentType</code>	<code>Integer</code>	The Type of the inline component. Only one value (<code>MiniToc</code>) is currently supported..
<code>InsertLinks</code>	<code>Integer</code>	True to insert hyperlinks, False otherwise..
<code>NextInlineComponentInDoc</code> (<i>Read-Only</i>)	<code>Object</code>	Next inline component (<code>InlinComponent</code> Object).
<code>TextRange</code> (<i>Read-Only</i>)	<code>TextLoc</code>	Text range of the component
<code>Unique</code> (<i>Read-Only</i>)	<code>Integer</code>	The Inline component's UID.

Markers

An `Marker` object represents a marker.

Object name - Marker

`Marker` objects have the following properties.

Table 202: Marker Properties (Page 1 of 2)

Property Name	Data Type	Description
<code>Element</code>	<code>Object</code>	If the marker is a structured marker in the FM+SGML document, this is the <code>Element</code> object containing the marker.
<code>MarkerText</code>	<code>String</code>	The marker's text string.
<code>MarkerType</code>	<code>Integer</code>	FrameMaker Versions before 5.5. The current marker's type number. The <code>Session</code> property <code>MarkerNames</code> is a list of marker names. The marker type is an index into this list minus 1. The first marker type is number 0.
<code>MarkerTypeId</code>	<code>Object</code>	FrameMaker Versions 5.5 and greater. The current marker's Object type (<code>MarkerType</code>).

Table 202: Marker Properties (Page 2 of 2)

Property Name	Data Type	Description
NextMarkerInDoc (<i>Read-Only</i>)	Object	Next marker (Marker Object).
TextLoc (<i>Read-Only</i>)	TextLoc	Text location of the marker's symbol.

Marker types

A `MarkerType` object represents a marker type. Marker types are stored in the document.

Object name - `MarkerType`

`MarkerType` objects have the following properties.

Table 203: Marker Type Properties

Property Name	Data Type	Description
NextMarkerTypeInDoc	Object	Next marker type (<code>MarkerType</code> Object)
Name	String	The name of this marker type, as it appears in the user interface.
InvariantName	String	An internal name for the marker type. By default, this is the same as <code>Name</code> . However, this can differ from <code>Name</code> if the user interface is in another language.
OldTypeNum	Integer	A number to map markers from documents earlier than version 5.5 to this marker type. For example, assume the name of a marker type is <code>MyMarkerType</code> , and the <code>OldTypeNum</code> is 11. Then markers of type 11 from earlier documents will import as markers of type <code>MyMarkerType</code> .
Public	Integer	True if the marker type should appear in the user interface. The default is <code>True</code> .
Transient	Integer	True if markers of this type should not be saved to files. The default is <code>False</code> .
Required (<i>Read-Only</i>)	Integer	True if the marker type is required by FrameMaker. The default is <code>False</code> .

Menus, Commands, and Menu Item Separators

The `Menu` objects are to represent menus, the `Command` object is used to represent menu commands, and `MenuItemSeparator` objects are used to represent menu item separators. The Menus, commands and menu item separators may be FrameMaker or User defined.

Object name - Menu

An `Menu` object represents a menu. `Menu` objects have the following properties.

Table 204: Menu Properties

Property Name	Data Type	Description
MenuType (Read-Only)	Integer	Type of menu: MenuMenuBar: a menu bar defined by the FrameMaker MenuPopup: a pop-up menu MenuAdHocRuler: an ad hoc formatting menu that appears on the ruler MenuDefault: a pull-down or pull-right menu
Name (Read-Only)	String	The command, menu, or menu item separator name. ^a
NextMenuItemInSession (Read-Only)	Object	The next menu item, menu, or separator in the list of menu items, menus, and separators in the session.
FirstMenuItemInMenu	Object	The first menu item or menu in the menu.
Label	String	The label the user sees on a menu. The label for menu item separators is read-only; it is always ---.
MenuItemIsEnabled (Read-Only)	Integer	True if the menu or menu item is enabled or False if it is disabled (dimmed).
NextMenuItemInMenu	Object	The next menu item, menu, or separator in the menu.
PrevMenuItemInMenu	Object	The previous menu item, menu, or separator in the menu.

a. The names for the default, predefined separators are !Separator, !Separator1, !Separator2, !Separator3, !Separator4, and !Separator5.

Object name - Command

An `Command` object represents a command or a menu item (a command that appears on a menu). `Command` objects have the following properties.

Table 205: Menu Command Properties (Page 1 of 3)

Property Name	Data Type	Description
AlwaysInstall (Read-Only)	Integer	If the command is for an installed ElmScript script, this property is True if the script is installed automatically at start up, False otherwise. It returns False, if the command is not related to an installed ElmScript script.
CmdSrcType (Read-Only)	String	This property returns 'Event' if the command is for an installed ElmScript event script and 'Standard' if the command is for an installed ElmScript standard script. If the command is not related to an installed ElmScript script it returns 'Other'.

Table 205: Menu Command Properties (Page 2 of 3)

Property Name	Data Type	Description
Description	String	A description of the command. FM 2017 or greater.
Fcode (Read-Only)	Code	An f-code that FrameMaker executes when the user chooses the menu item or presses the keyboard shortcut.
Fcodes (Read-Only)	UIntList	The list of f-codes that FrameMaker executes when the user chooses the menu item or presses the keyboard shortcut. Normally, the first f-code in the list is the same as the f-code specified by Fcode .
MenuItemIsEnabled (Read-Only)	Integer	True if the menu or menu item is enabled or False if it is disabled (dimmed).
MenuItemType (Read-Only)	Integer	The type of command or menu item: MenuItemFrame : the command is a menu item defined by the FrameMaker. MenuItemApi : the command is a menu item defined by a ElmScript script. MenuItemMacro : the menu item is not a command; it calls a macro. MenuItemExpandOMatic : the menu item is an expandomatic menu item (such as !ShowParagraphTags) defined by the FrameMaker.
Mode (Read-Only)	Integer	The mode in which keyboard shortcuts are recorded: ModeAll ModeMath ModeNonMath
Name (Read-Only)	String	The command, menu, or menu item separator name. ^a
CanHaveCheckMark	Integer	True if the menu item can have a check mark. If the menu item is defined by FrameMaker, you can get this property, but not set it.
CheckMarkIsOn	Integer	True if the menu item can have a check mark and the check mark is on. If the menu item is defined by FrameMaker, you can get this property, but not set it.
EnabledWhen	Integer	The context in which the menu item is enabled. For a list of the constants that this field can specify, see “EnabledWhen Context Descriptions” on page 244. If the menu item is defined by FrameMaker, you can get this property, but not set it.
ExpandOMaticParent (Read-Only)	Object	If the menu item is an expandomatic menu item, its virtual parent Object(Command) .
HasShiftOrUnshiftCommand	Integer	Specifies whether a command has an accompanying shift command or unshift command: ItemHasShiftCommand ItemHasUnshiftCommand ItemHasNoShiftOrUnshiftCommand
HelpLink	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.doc:lastpage</code> . Do not specify hypertext commands such as <code>gotopage</code> . The FrameMaker automatically prepends the appropriate hypertext command to the HelpLink string when the user requests context-sensitive help. If the destination file is not in the client directory, the FrameMaker looks for it in the FrameMaker help directory. This property is valid only for commands created by ElmScript scripts. It is not valid for commands created directly by FrameMaker.
KeyboardShortcutLabel	String	The keyboard shortcut string that appears on the menu. This string need not be one of the actual shortcuts specified by KeyboardShortcuts .

Table 205: Menu Command Properties (Page 3 of 3)

Property Name	Data Type	Description
KeyboardShortcuts	StringList	The list of keyboard shortcuts the user can press to execute the command. To add a shortcut, append it to the list. You cannot delete shortcuts from the list.
Label	String	The label the user sees on a menu. The label for menu item separators is read-only; it is always ---.
Labels	StringList	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, <code>Labels</code> specifies only the string for that label. If the menu item has different labels in different contexts, <code>Labels</code> specifies pairs of strings with the following format: <i>Context,Label</i> where <i>Label</i> specifies the menu item label and <i>Context</i> specifies the context in which the label appears on the menu.
NextCommandInSession (Read-Only)	Object	The next command in the list of commands in the session.
NextMenuItemInMenu	Object	The next menu item, menu, or separator in the menu.
NextMenuItemInSession (Read-Only)	Object	The next menu item, menu, or separator in the list of menu items, menus, and separators in the session.
PrevMenuItemInMenu	Object	The previous menu item, menu, or separator in the menu.
ScriptFileName (Read-Only)	String	If the command is for an installed ElmScript script, this property returns the name of the file for that script. If the command is not related to an installed ElmScript script it returns <code>Null</code> .
ScriptName (Read-Only)	String	If the command is for an installed ElmScript script, this property returns the name of the script. If the command is not related to an installed ElmScript script it returns <code>Null</code> .
ShiftOrUnshiftCommand	Object	If <code>HasShiftOrUnshiftCommand</code> is set to <code>ItemHasShiftCommand</code> , the command object to use when the user holds down the Shift key. If <code>HasShiftOrUnshiftCommand</code> is set to <code>ItemHasUnshiftCommand</code> , the command object to use when the user isn't holding down the Shift key.

a. The names for the default, predefined separators are `!Separator`, `!Separator1`, `!Separator2`, `!Separator3`, `!Separator4`, and `!Separator5`.

The following table lists the values `EnabledWhen` can have and the corresponding contexts in which a menu item is active

Table 206: EnabledWhen Context Descriptions (Page 1 of 2)

EnabledWhen value	Command Context Description
EnableAlwaysDisable	No context. The menu item is disabled. If a menu item is enabled and you set <code>EnabledWhen</code> to this value, it disables and dims the menu item.
EnableAlwaysEnable	All contexts. This is the default value. If the menu item is disabled, setting <code>EnabledWhen</code> to this value enables it.
EnableCanPaste	The Clipboard contains an object or text that can be pasted at the insertion point.
EnableCopy	Some text or an object is selected.
EnableCopyFont	The insertion point or selection is in the text of a paragraph, a math object, a table, or a text line.
EnableInCellText	The insertion point or selection is in a table cell.
EnableInFlow	A text frame is selected, or the insertion point or selection is in a paragraph.

Table 206: EnabledWhen Context Descriptions (Page 2 of 2)

EnabledWhen value	Command Context Description
EnableInMath	The insertion point or selection is in a math object.
EnableInParaText	The insertion point or selection is in a paragraph (but not in a math object).
EnableInTable	The insertion point or selection is in any part of a table.
EnableInTableTitle	The insertion point or selection is in the table title.
EnableInText	The insertion point or selection is in a graphic text line or a paragraph.
EnableInTextLine	The insertion point or selection is in a graphic text line.
EnableIsAFrame	The first selected object is an anchored frame.
EnableIsCell	A single cell in a table is selected.
EnableIsCells	One or more cells in a table are selected.
EnableIsGraphicInset	The first selected object is a graphic inset.
EnableIsObj	An object is selected.
EnableIsOrInFrame	The selected object is a graphic frame or is in a graphic frame that is not a page frame.
EnableIsTable	An entire table is selected.
EnableIsTextFrame	A text frame is selected.
EnableIsTextInset	The first selected object is a text inset.
EnableIsTextSel	The selection is in a paragraph.
EnableIsViewOnly	The current document is locked.
EnableNeedsBookpOnly	A book is open.
EnableNeedsDocpOnly	A document is open.
EnableObjProps	The insertion point is in text, a table, or a math object, or a graphic object is selected.
EnableNeedsDocPOrBookP	A document or a book is open.
EnableBookHasSelection	The book has a selection made.
EnableDocOrBookHasSelection	A document is in front or a book has a selection.

Object name - MenuItemSeparator

An MenuItemSeparator object represents a menu item separator.

Table 207: Menu Item Separator Properties (Page 1 of 2)

Property Name	Data Type	Description
MenuItemIsEnabled (Read-Only)	Integer	True if the menu or menu item is enabled or False if it is disabled (dimmed).
Name (Read-Only)	String	The command, menu, or menu item separator name. ^a
NextMenuItemInSession (Read-Only)	Object	The next menu item, menu, or separator in the list of menu items, menus, and separators in the session.

Table 207: Menu Item Separator Properties (Page 2 of 2)

Property Name	Data Type	Description
Label	String	The label the user sees on a menu. The label for menu item separators is read-only; it is always ---.
NextMenuItemInMenu	Object	The next menu item, menu, or separator in the menu.
PrevMenuItemInMenu	Object	The previous menu item, menu, or separator in the menu.

a. The names for the default, predefined separators are !Separator, !Separator1, !Separator2, !Separator3, !Separator4, and !Separator5.

Pages

There are four types of pages. The `BodyPage`, `HiddenPage`, `MasterPage`, and `RefPage` objects are used to represent them.

Object name - `BodyPage`

An `BodyPage` object represents a body page. `BodyPage` objects have the following properties.

Table 208: Body Page Properties (Page 1 of 2)

Property Name	Data Type	Description
MasterPage	String	Name of master page background for body page if <code>PageBackground</code> is set to <code>BgdOther</code> . It is <code>ZERO</code> if <code>PageBackground</code> is set to <code>BgdDefault</code> or <code>BgdNone</code> .
PageBackground	Integer	Type of master page background: <code>BgdDefault</code> : 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. ^a <code>BgdNone</code> : The page has no master page background. <code>BgdOther</code> : The page has the custom master page background specified by <code>MasterPage</code> .
PageFrame (<i>Read-Only</i>)	Object	The page's page frame (<code>UnanchoredFrame</code> Object).
PageHeight (<i>Read-Only</i>)	Metric	Height of page.
PageIsRecto (<i>Read-Only</i>)	Integer	True if right body page or <code>False</code> if left body page.
PageNext (<i>Read-Only</i>)	Object	Next body page (<code>BodyPage</code> Object) in the document.
PageNum (<i>Read-Only</i>)	Integer	Page number.
PageNumString (<i>Read-Only</i>)	String	Page number string.
PagePrev (<i>Read-Only</i>)	Object	Previous body page (<code>BodyPage</code> Object) in the document.

Table 208: Body Page Properties (Page 2 of 2)

Property Name	Data Type	Description
PageWidth (Read-Only)	Metric	Width of page.
PointPageNum (Read-Only)	Integer	Number of point page.

a. To determine if a body page has a Left or a Right master page background when its `PageBackground` property is set to `BgdDefault`, query its `PageIsRecto` property.

Object name - HiddenPage

An `HiddenPage` object represents a hidden page. `HiddenPage` objects have the following properties.

Table 209: Hidden Page Properties

Property Name	Data Type	Description
Name (Read-Only)	String	Name of hidden page
PageFrame (Read-Only)	Object	Page frame (<code>UnanchoredFrame</code> Object)
PageHeight (Read-Only)	Metric	Height of page
PageWidth (Read-Only)	Metric	Width of page

Object name - MasterPage

An `MasterPage` object represents a master page. `MasterPage` objects have the following properties.

Table 210: Master Page Properties

Property Name	Data Type	Description
Name	String	Name of master page (for example, <code>Right</code> or <code>Left</code>)
PageFrame (Read-Only)	Object	Page frame (<code>UnanchoredFrame</code> Object)
PageHeight (Read-Only)	Metric	Height of page
PageNext (Read-Only)	Object	Next master page (<code>MasterPage</code> Object) in the document
PageNum (Read-Only)	Integer	Page number
PagePrev (Read-Only)	Object	Previous master page (<code>MasterPage</code> Object) in the document
PageWidth (Read-Only)	Metric	Width of page

Object name - RefPage

An `RefPage` object represents a reference page. `RefPage` objects have the following properties.

Table 211: Reference Page Properties

Property Name	Type	Description
Name	String	Name of reference page
PageFrame (<i>Read-Only</i>)	Object	Page frame (<code>UnanchoredFrame</code> Object)
PageHeight (<i>Read-Only</i>)	Metric	Height of page
PageNext (<i>Read-Only</i>)	Object	Next reference page (<code>RefPage</code> Object) in the document
PageNum (<i>Read-Only</i>)	Integer	Page number
PagePrev (<i>Read-Only</i>)	Object	Previous reference page (<code>RefPage</code> Object) in the document
PageWidth (<i>Read-Only</i>)	Metric	Width of page

Paragraphs

The `Pgf` object is used to represent a paragraph instance and an `PgfFmt` object is used to represent a paragraph format in the Paragraph Catalog. `Pgf` and `PgfFmt` objects have many of the same properties.

Object name - Pgf

`Pgf` objects have the following properties. A paragraph is the object where FrameMaker stores most of the text in a document. Flows, Textframes, Footnotes, Table titles, and table cells store their text in paragraphs. The exception is the Textline object, included under the graphic object heading.

Table 212: Paragraph Properties (Page 1 of 6)

Property Name	Data Type	Description
AdjHyphens	Integer	Number of allowable adjacent hyphens
AutoNumChar	String	Character Format for the automatic numbering string specified by <code>AutoNumString</code> ; ' ' if the default character format is used
AutoNumString	String	Autonumber Format string; for example, "<n>.<n+>"
BkColor	Object	Background color object(<code>Color</code> Object). FM 10 or greater .
BlockLines	Integer	The number of Widow/Orphan lines
BottomSeparator	String	Name of frame to put below paragraph

Table 212: Paragraph Properties (Page 2 of 6)

Property Name	Data Type	Description
Capitalization	Integer	Type of capitalization to use: CapitalCaseNorm CapitalCaseSmall CapitalCaseLower CapitalCaseUpper
CellBottomMargin	Metric	Amount added to default bottom margin of table cell.
CellLeftMargin	Metric	Amount added to default left margin of table cell.
CellMarginsFixed	Integer	Specifies which cell margins are fixed. The following values can be ORed into it: PgFixedBMargin: the bottom margin is fixed PgFixedLMargin: the left margin is fixed PgFixedRMargin: the right margin is fixed PgFixedTMargin: 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 PgFixedBMargin is set, CellBottomMargin specifies the absolute value of the cell's bottom margin, overriding the cell margin specified by the table format. If PgFixedBMargin is not set, CellBottomMargin is added to the margin specified by the table format.
CellRightMargin	Metric	Amount added to default right margin of table cell.
CellTopMargin	Metric	Amount added to default top margin of table cell.
CellVAlignment	Integer	Vertical alignment of a paragraph when it is the first one in a cell: PgVAlignTop PgVAlignMiddle PgVAlignBottom
ChangeBar	Integer	True if Change Bars are on.
Color	Object	Spot color (Color Object).
CombinedFontFamily	Object	Combined font definition (CombinedFontDefn)
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
FirstIndent	Metric	First-line left margin, measured from left side of current text column (0 cm to 100 cm)
FontAngle	Integer	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName (Read-Only)	String	The font's encoding
FontFamily	Integer	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).
FontPlatformName	String	Name that uniquely identifies a font on a specific platform.

Table 212: Paragraph Properties (Page 3 of 6)

Property Name	Data Type	Description
FontPostScriptName	String	Name given to a font when it is sent to a PostScript printer.
FontSize	Metric	Font size (2 pt to 400 pt).
FontVariation	Integer	Font variation (specifies an index into the array of font variations provided by the session property <code>FontVariationNames</code>).
FontWeight	Integer	Font weight (specifies an index into the array of font weights provided by the session property <code>FontWeightNames</code>).
FormatOverride	Integer	<code>True</code> if the paragraph contains a paragraph format override.
Hyphenate	Integer	<code>True</code> if Automatic Hyphenation is enabled
HyphMinPrefix	Integer	Minimum number of letters that must precede hyphen
HyphMinSuffix	Integer	Minimum number of letters that must follow a hyphen
HyphMinWord	Integer	Minimum length of a hyphenated word
InTextFrame (Read-Only)	Object	Text frame containing the paragraph (<code>TextFrame</code> Object)
InTextObj (Read-Only)	Object	Subcolumn, footnote, or table cell the paragraph begins in (<code>SubCol</code> , <code>Fn</code> , or <code>Cell</code> Object)
KeepWithNext	Integer	<code>True</code> if Keep With Next Paragraph is enabled
KeepWithPrev	Integer	<code>True</code> if Keep With Previous Paragraph is enabled
KernX	Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Language	Integer	Hyphenation and spell-checking language to use: LangBrazilian LangBritish LangCanadianFrench LangCatalan LangDanish LangDutch LangEnglish LangFinnish LangFrench LangGerman LangItalian LangNewDutch LangNewGerman LangNewSwissGerman LangNoLanguage LangNorwegian LangNynorsk LangPortuguese LangSpanish LangSwedish LangSwissGerman LangJapanese LangTraditionalChinese LangSimplifiedChinese LangKorean and FM 8.0 or greater LangGreek LangRussian LangCzech LangPolish LangHungarian LangTurkish LangSlovak LangSlovenian LangBulgarian LangCroatian LangEstonian LangLatvian LangLithuanian LangRomanian and FM 2015 or greater LangRomanian LangArabic LangHebrew and FM 2017 or greater LangThai LangFarsi
Leading	Metric	Space below each line in a paragraph
LeftIndent	Metric	Left margin, measured from left side of current text column (0 cm to 100 cm)
LetterSpace	Integer	<code>True</code> if Word Spacing is enabled

Table 212: Paragraph Properties (Page 4 of 6)

Property Name	Data Type	Description
LineSpacing	Integer	Space between lines in a paragraph measured from baseline to baseline: PgFixed: default font size PgProportional: largest font in line PgFloating: largest ascender in line
Locked	Integer	True if the paragraph is part of a text inset that retains formatting information from the source document. The paragraph is not affected by global formatting performed on the document.
MaxJLetSpace	Metric	Maximum Asian letter space
MaxJRomSpace	Metric	Maximum Asian-Roman space
MaxSpace	Metric	Maximum word spacing (percentage of an em space in current font)
MinJLetSpace	Metric	Minimum Asian letter space
MinJRomSpace	Metric	Minimum Asian-Roman space
MinSpace	Metric	Minimum word spacing (percentage of an em space in current font)
Name	String	Name of paragraph format
NextPgInDoc (Read-Only)	Object	Next paragraph in the document (Pgf Object)
NextPgInFlow (Read-Only)	Object	Next paragraph in the flow (Pgf Object)
NextTag	String	Tag for new next paragraph
NumAtEnd	Integer	True if numbering position is End of Paragraph; False if it is Beginning of Paragraph
NumTabs (Read-Only)	Integer	Number of tabs in the paragraph.
OptJLetSpace	Metric	Optimum Asian letter space
OptJRomSpace	Metric	Optimum Asian-Roman space
OptSpace	Metric	Optimum word spacing
Outline	Integer	True if Outline is enabled (Macintosh only).
Overline	Integer	True if Overline is enabled.
PairKern	Integer	True if Pair Kern is enabled.
PDFStructureLevel	Integer	The PDF structure level of paragraphs with the current format. This property is used when the PDFStructure property is True for the document, and FrameMaker generates PDF data. The value for this property can be between 0 and 100, where greater values are deeper in the hierarchy. If PDFStructureLevel is 0, FrameMaker does not include paragraphs of this format in the PDF structure.
PgfAlignment	Integer	Horizontal alignment of paragraph: PgfLeft PgfRight PgfCenter PgfJustified
PgfBoxColor	Object	(Color Object). FM 12 or greater
PgfIsAutoNum	Integer	True if autonumbering is enabled

Table 212: Paragraph Properties (Page 5 of 6)

Property Name	Data Type	Description
PgfNumber (<i>Read-Only</i>)	String	The formatted string representation of the paragraph number; for example, 1.2 for a paragraph whose <code>AutoNumString</code> property is set to <code><n>.<n+></code>
PgfSpellChecked	Integer	True if paragraph has been spell-checked
Placement	Integer	Paragraph placement: PgfSideBody PgfSideHeadTop PgfSideHeadFirstBaseline PgfSideHeadLastBaseline PgfRunIn PgfStraddle PgfStraddleNormalOnly
Position	Integer	Specifies position relative to baseline of text: <code>PosNorm</code> : Normal <code>PosSub</code> : Subscript <code>PosSuper</code> : Superscript
PrevPgfInFlow (<i>Read-Only</i>)	Object	Previous paragraph in the flow (<code>Pgf</code> Object)
RightIndent	Metric	Right margin, measured from right side of current text column
RunInSeparator	String	String for Run-In Head Default Punctuation
Shadow	Integer	True if Shadow is enabled (Macintosh only).
SpaceAbove	Metric	Space above paragraph
SpaceBelow	Metric	Space below paragraph
Spread	Metric	FrameMaker 5.1x only. FrameMaker 5.5 use the "tracking" property.
Start	Integer	Vertical placement of paragraph: PgfAnywhere PgfTopOfCol PgfTopOfPage PgfTopOfLeftPage PgfTopOfRightPage
Stretch	Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	Integer	True if Strikethrough is enabled.
Tabs	TabList	Array of tab descriptions that specify the positions and types of tab stops.
Text (<i>Read-Only</i>)	String	Text of the paragraph.
TextRange	TextRange	The <code>TextRange</code> encompassing the paragraph.
TopSeparator	String	Name of frame to put above paragraph

Table 212: Paragraph Properties (Page 6 of 6)

Property Name	Data Type	Description
Underlining	Integer	Type of underlining: CbNoUnderline CbSingleUnderline CbDoubleUnderline CbNumericUnderline
UseNextTag	Integer	True if Next Paragraph Tag is enabled
YakumonoType	Integer	The Yakumono rules to handle punctuation characters; can be one of FloatingYakumono MonospaceYakumono FixedYakumono

Object name - PgfFmt

PgfFmt objects (paragraph formats in the Paragraph Catalog) have the following properties.

Table 213: Paragraph Format Properties (Page 1 of 5)

Property Name	Data Type	Description
AcrobatLevel	Integer	The outline level of paragraphs with the format when the FrameMaker generates Adobe Acrobat data. If AcrobatLevel is 0, FrameMaker does not generate bookmarks for paragraphs with the format. If AcrobatLevel is greater than 0, FrameMaker sets the outline level of paragraphs with this format to the specified level.
AdjHyphens	Integer	Number of allowable adjacent hyphens
AutoNumChar	String	Character format for the automatic numbering string specified by AutoNumString ; ' ' if the default character format is used
AutoNumString	String	Autonumber format string (for example, <n>.<n+>)
BkColor	Object	Background color object(Color Object). FM 10 or greater .
BlockLines	Integer	The number of Widow/Orphan lines
BottomSeparator	String	Name of frame to put below paragraph
Capitalization	Integer	Type of capitalization to use: CapitalCaseNorm CapitalCaseSmall CapitalCaseLower CapitalCaseUpper
CellBottomMargin	Metric	Amount added to table cell default bottom margin
CellLeftMargin	Metric	Amount added to table cell default left margin

Table 213: Paragraph Format Properties (Page 2 of 5)

Property Name	Data Type	Description
CellMarginsFixed	Integer	Specifies which cell margins are added to default table cell margins. The following values can be ORed into it. PgFixedBMargin: the bottom margin is added PgFixedLMargin: the left margin is added PgFixedRMargin: the right margin is added PgFixedTMargin: the top margin is added
CellRightMargin	Metric	Amount added to table cell default right margin
CellTopMargin	Metric	Amount added to table cell default top margin
CellVAlignment	Integer	Vertical alignment of a paragraph when it is the first one in a cell: PgVAlignTop PgVAlignMiddle PgVAlignBottom
ChangeBar	Integer	True if Change Bars is on.
Color	Object	Spot color (Color Object).
CombinedFontFamily	Object	Combined font definition (CombinedFontDefn)
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
FirstIndent	Metric	First-line left margin, measured from left side of current text column (0 pt to 360 pt)
FontAngle	Integer	Font angle (specifies an index into the array of font angles provided by the session property FontAngleNames).
FontEncodingName (Read-Only)	String	The font's encoding
FontFamily	Integer	Font family (specifies an index into the array of font families provided by the session property FontFamilyNames).
FontPlatformName	String	Name that uniquely identifies a font on a specific platform.
FontPostScriptName	String	Name given to a font when it is sent to a PostScript printer.
FontSize	Metric	Font size (2 pt to 400 pt).
FontVariation	Integer	Font variation (specifies an index into the array of font variations provided by the session property FontVariationNames).
FontWeight	Integer	Font weight (specifies an index into the array of font weights provided by the session property FontWeightNames).
Hyphenate	Integer	True if Hyphenation is enabled
HyphMinPrefix	Integer	Minimum number of letters that must precede hyphen
HyphMinSuffix	Integer	Minimum number of letters that must follow a hyphen

Table 213: Paragraph Format Properties (Page 3 of 5)

Property Name	Data Type	Description
HyphMinWord	Integer	Minimum length of a hyphenated word
KeepWithNext	Integer	True if Keep With Next Paragraph is enabled
KeepWithPrev	Integer	True if Keep With Previous Paragraph is enabled
KernX	Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right and a negative value moves a character left.
KernY	Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up and a negative value moves characters down.
Language	Integer	Hyphenation and spell-checking language to use: LangBrazilian LangBritish LangCanadianFrench LangCatalan LangDanish LangDutch LangEnglish LangFinnish LangFrench LangGerman LangItalian LangNewDutch LangNewGerman LangNewSwissGerman LangNoLanguage LangNorwegian LangNynorsk LangPortuguese LangSpanish LangSwedish LangSwissGerman LangJapanese LangTraditionalChinese LangSimplifiedChinese LangKorean and FM 8.0 or greater LangGreek LangRussian LangCzech LangPolish LangHungarian LangTurkish LangSlovak LangSlovenian LangBulgarian LangCroatian LangEstonian LangLatvian LangLithuanian LangRomanian and FM 2015 or greater LangRomanian LangArabic LangHebrew
Leading	Metric	Space below each line in a paragraph
LeftIndent	Metric	Left margin, measured from left side of current text column (0 pt to 360 pt)
LetterSpace	Integer	True if Word Spacing is enabled
LineSpacing	Integer	Space between lines in a paragraph measured from baseline to baseline: PgFixed: default font size PgProportional: largest font in line PgFloating: largest ascender in line
MaxJLetSpace	Metric	Maximum Asian letter space
MaxJRomSpace	Metric	Maximum Asian-Roman space
MaxSpace	Metric	Maximum word spacing (percentage of an em space in current font)
MinJLetSpace	Metric	Minimum Asian letter space
MinJRomSpace	Metric	Minimum Asian-Roman space
MinSpace	Metric	Minimum word spacing (percentage of an em space in current font)
Name	String	Name of paragraph format
NextPgFmtInDoc (Read-Only)	Object	Next paragraph format in document (PgFmt Object)
NextTag	String	Tag for following paragraph
NumAtEnd	Integer	True if numbering position is End of Paragraph; False if it is Beginning of Paragraph
NumTabs (Read-Only)	Integer	Number of tabs in the paragraph

Table 213: Paragraph Format Properties (Page 4 of 5)

Property Name	Data Type	Description
OptJLetSpace	Metric	Optimum Asian letter space
OptJRomSpace	Metric	Optimum Asian-Roman space
OptSpace	Metric	Optimum word spacing
Outline	Integer	True if Outline is enabled (Macintosh only).
Overline	Integer	True if Overline is enabled.
PairKern	Integer	True if Pair Kern is enabled.
PDFStructureLevel	Integer	The PDF structure level of paragraphs with the current format. This property is used when the PDFStructure property is True for the document, and FrameMaker generates PDF data. The value for this property can be between 0 and 100, where greater values are deeper in the hierarchy. If PDFStructureLevel is 0, FrameMaker does not include paragraphs of this format in the PDF structure.
PgfAlignment	Integer	Horizontal alignment of paragraph: PgfLeft PgfRight PgfCenter PgfJustified
PgfBoxColor	Object	(Color Object). FM 12 or greater
PgfIsAutoNum	Integer	True if autonumbering is enabled
Placement	Integer	Paragraph placement: PgfSideBody PgfSideHeadTop PgfSideHeadFirstBaseline PgfSideHeadLastBaseline PgfRunIn PgfStraddle PgfStraddleNormalOnly
Position	Integer	Specifies position relative to baseline of text: PosNorm: Normal PosSub: Subscript PosSuper: Superscript
RightIndent	Metric	Right margin, measured from right side of current text column
RunInSeparator	String	String for Run-In Head Default Punctuation
Shadow	Integer	True if Shadow is enabled (Macintosh only).
SpaceAbove	Metric	Space before paragraph
SpaceBelow	Metric	Space after paragraph
Spread	Metric	Character spread expressed as a percentage of an em (metric –100% to 1000%).

Table 213: Paragraph Format Properties (Page 5 of 5)

Property Name	Data Type	Description
Start	Integer	Vertical placement of paragraph: PgfAnywhere PgfTopOfCol PgfTopOfPage PgfTopOfLeftPage PgfTopOfRightPage
Stretch	Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric –10% to 1000%).
Strikethrough	Integer	True if Strikethrough is enabled.
Tabs	TabList	Array of tab descriptions that specify the positions and types of tab stops
TopSeparator	String	Name of frame to put above paragraph
Tsume	Integer	True if Tsume is enabled. Select Tsume to move a Japanese character closer to the characters next to it. This option is available only when you can type Japanese text in documents and dialog boxes. The amount of space a variable-width character can move is determined by the metrics for that character.
Underlining	Integer	Type of underlining: CbNoUnderline CbSingleUnderline CbDoubleUnderline CbNumericUnderline
UseNextTag	Integer	True if Next Paragraph Tag is enabled
YakumonoType	Integer	The Yakumono rules to handle punctuation characters; can be one of FloatingYakumono MonospaceYakumono FixedYakumono

Rubi composites

An `Rubi` object represents a rubi composite.

Object name - Rubi

`Rubi` objects have the following properties.

Table 214: Rubi Properties (Page 1 of 2)

Property Name	Type	Description
<code>Element†</code>	Object	If the rubi group is in a structured document, the object of the associated <code>Element</code> for the rubi group element.
<code>OyamojiTextRange (Read-Only)</code>	<code>TextRange</code>	The text range the oyamoji text encompasses.

Table 214: Rubi Properties (Page 2 of 2)

Property Name	Type	Description
NextRubiInDoc (<i>Read-Only</i>)	TextRange	The next instance of a rubi composite (Rubi Object) in the document..
RubiElement†	Object	If the rubi group is in a structured document, the object handle of the associated Element for the rubi element.
RubiTextRange (<i>Read-Only</i>)	TextRange	The text range the rubi text encompasses.
Unique (<i>Read-Only</i>)	Integer	The rubi composite's UID.

Session

The `Session` object is used to represent a FrameMaker session.

Object name - Session

`Session` objects have the following properties.

Table 215: Session Properties (Page 1 of 6)

Property Name	Data Type	Description
ActiveBook	Object	The book with input focus (<code>Book</code> Object).
ActiveDoc	Object	The document with input focus (<code>Doc</code> Object).
AddMarkerTypeToStandardMarkers	String	The name of a marker type to add to the standard list of marker types.
AllowNewFileURL	Integer	True/False. FM 10 or greater
ApplyFormatRules	Integer	True if element formatting is enabled.
AutoBackup	Integer	True if Automatic Backup is enabled.
AutoSave	Integer	True if Automatic Save is enabled.
AutoSaveSeconds	Integer	Time between automatic saves in seconds (60 seconds to 10800 seconds).
BkGndColor	Object	Background color object(<code>Color</code> Object). FM 10 or greater .
CTFontContext	Integer	Undocumented option.
CurrentMenuSet	Integer	Type of menu set: MenuQuick MenuComplete MenuCustom
DefaultFontAngle	Integer	Undocumented property.
DefaultFontFamily	Integer	Undocumented property.
DefaultFontVariation	Integer	Undocumented property.

Table 215: Session Properties (Page 2 of 6)

Property Name	Data Type	Description
DefaultFontWeight	Integer	Undocumented property.
DefaultKeyCatalog	Object	Undocumented property. FM 11 or greater
DefaultVectorFormatForXMLExport	Object	Undocumented property. FM 11 or greater
Displaying	Integer	False if screen refresh is completely turned off.
DisplayName (Read-Only)	String	The name of the display on which FrameMaker session is running (UNIX only).
DitaMode	Integer	True/False. FM 10 or greater
DontLoadStartupClients	StringList	FM 12 or greater.
DoNotExportInvalidXML	Integer	True/False. FM 10 or greater
DoPostXSLTValidationOnExport	Integer	True/False. FM 10 or greater
EnableAutoSpellCheck	Integer	True if the auto spell check is enabled. FM 10 or greater
ExportFilters (Read-Only)	StringList	List of export filter strings. Characters 9 through 12 are the filter format id.
FirstCommandInSession	Object	First command in the list of commands in the session (Command Object).
FirstKeyCatalogInSession (Read-Only)	Object	Undocumented property.
FirstMenuItemInSession (Read-Only)	Object	First menu item or menu in the list of menus, menu items, and menu item separators in the session (Command, Menu, MenuItemSeparator Object).
FirstOpenBook (Read-Only)	Object	First open book (Book Object) in session.
FirstOpenDoc (Read-Only)	Object	First open document (Doc Object) in session.
FMBinDir (Read-Only)	String	Directory pathname of \$FMHOME/fmunit/bin (on UNIX or Windows) or the Modules folder (on the Macintosh).
FMConsoleString (Read-Only)	String	Undocumented property. FM 11 or greater
FMCurrentDir (Read-Only)	String	Name of the directory from which FrameMaker was started (on UNIX or Windows) or the Modules folder (on the Macintosh).
FMHelpDir (Read-Only)	String	Pathname of the FrameMaker help directory
FMHomeDir (Read-Only)	String	Pathname of \$FMHOME (on UNIX or Windows) or the FrameMaker folder (on the Macintosh).
FMInitDir (Read-Only)	String	Directory pathname of \$FMHOME/fmunit (on UNIX or Windows) or the Modules folder (on the Macintosh).
FMSgmlDir (Read-Only)	String	Directory pathname of \$FMHOME/structure./sgml
FMStructureDir (Read-Only)	String	Directory pathname of \$FMHOME/structure. FM 7.0 or greater
FMXmlDir (Read-Only)	String	Directory pathname of \$FMHOME/structure./xml

Table 215: Session Properties (Page 3 of 6)

Property Name	Data Type	Description
FontAngleNames (Read-Only)	StringList	List of font angles available in the current session. FM 7.0 or greater
FontFamilyAttributes (Read-Only)	IntList	An array of flags that indicate attributes for each font family listed by FontFamilyNames . This array of integers is indexed the same as the list of font family names, and corresponds directly to that list. Each Integer 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, their mask values, and their meaning follow: FamilyVisible (0x00000001): Family is visible in menu. FamilySelectable (0x00000002): Family can be selected in menu. FamilyMapped (0x00000004): Family is is always mapped to another family. FamilySurrogate (0xFFFF0000): The family mapped to if FamilyMapped is True.
FontFamilyFullNames (Read-Only)	StringList	List of font family full names available in the current session. Note that this list does not include combined fonts.
FontFamilyNames (Read-Only)	StringList	List of font family names available in the current session. Note that this list does not include combined fonts.
FontVariationNames (Read-Only)	StringList	List of font variations available in the current session.
FontWeightNames (Read-Only)	StringList	List of font weights available in the current session.
HostName (Read-Only)	String	Name of the host computer.
Gravity	Integer	True if gravity is turned on. FM 6.0 or greater
GreekSize	Metric	Size for Greek text. FM 6.0 or greater
IconBarOn	Integer	True if the four icons that appear on the upper-right side of the document window are on. Changing this property affects only documents that are opened subsequently; it does not affect documents that are already open.
ImportFilters (Read-Only)	StringList	List of export filter strings. Characters 9 through 12 are the filter format id.
InTableSortOperation	Integer	True/False. FM 10 or greater
IsFMRunningAsServer	Integer	True if the FrameMaker is running as a server. FM 10 or greater
IsIconified	Integer	True if the FrameMaker window is iconified (UNIX and Windows only).
IsInFront	Integer	True if the FrameMaker window is in front of other application windows. You can use this property to bring the FrameMaker to the front or back. However, reading this property on Unix is not always reliable because the XServer doesn't necessarily update the property immediately.
IsOnScreen	Integer	True if the FrameMaker window is visible on the screen (UNIX only).
IsTempOpenSave	Integer	Undocumented property. FM 11 or greater
KeyCatalogWorkflow	Integer	Undocumented property. FM 11 or greater
Label	String	The title in the FrameMaker window title bar (UNIX and Windows only).

Table 215: Session Properties (Page 4 of 6)

Property Name	Data Type	Description
Language (Read-Only)	Integer	Product language: LangBrazilian LangBritish LangCanadianFrench LangCatalan LangDanish LangDutch LangEnglish LangFinnish LangFrench LangGerman LangItalian LangNewDutch LangNewGerman LangNewSwissGerman LangNoLanguage LangNorwegian LangNynorsk LangPortuguese LangSpanish LangSwedish LangSwissGerman LangJapanese LangTraditionalChinese LangSimplifiedChinese LangKorean and FM 8.0 or greater LangGreek LangRussian LangCzech LangPolish LangHungarian LangTurkish LangSlovak LangSlovenian LangBulgarian LangCroatian LangEstonian LangLatvian LangLithuanian LangRomanian and FM 2015 or greater LangRomanian LangArabic LangHebrew
LoadStartupScripts	Integer	Undocumented property. FM 12 or greater
MarkerNames (Read-Only)	StringList	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 are assigned to the document; use the MarkerTypeNames property of Doc to get the full list of marker types.
NoMultiMediaInPDF	Integer	True/False. FM 10 or greater
OpenDir	String	Directory pathname of last opened document or book (on UNIX) or the FrameMaker directory (on the Macintosh and Windows).
OperatingSystem (Read-Only)	String	Operating system under which the current session is running: Solaris SunOS HPUX AIX OSF1 IRIX Macintosh DOS
Path (Read-Only)	String	Pathname to search to start the FrameMaker (on UNIX or Windows) or the FrameMaker folder (on the Macintosh).
Platform (Read-Only)	String	Name of the platform on which the current session is running: Sun HP RS6000 Alpha SGI Macintosh Intel

Table 215: Session Properties (Page 5 of 6)

Property Name	Data Type	Description
ProductIsDemo	Integer	True if the current session is for a demo version of FrameMaker. FM 7.0 or greater
ProductIsStructured	Integer	True if FrameMaker is running in structured mode. FM 7.0 or greater
ProductName (Read-Only)	String	FrameMaker product name: FrameMaker Structured FrameMaker FrameViewer DemoMaker DemoMaker+SGML
ProgId (Read-Only)	String	FrameMaker product name: FM 10 or greater
ProcessNumber	Integer	The process number of the FrameMaker session (UNIX only).
RasterFilterDpi	Metric	Undocumented option.
Reformatting	Integer	True if reformatting is enabled.
RememberMissingFontNames	Integer	True if Remember Missing Font Names is activated .
RemoveExtraWhiteSpacesOnXMLImport	Integer	True/False. FM 10 or greater
RetainUndoState	Integer	Undocumented option.
RpcPropertyName	String	The RPC name of the FrameMaker session (UNIX only).
RpcProgramNumber	Integer	The RPC process number of the FrameMaker session (UNIX only).
ScreenHeight	Integer	Height of the FrameMaker window in pixels (UNIX and Windows only).
ScreenWidth	Integer	Width of the FrameMaker window in pixels (UNIX and Windows only).
ScreenX	Integer	The offset of the FrameMaker window in pixels from the left side of the screen (UNIX and Windows only). If you set a value that would result in the FrameMaker window being off the screen, that value is ignored and the old value is retained.
ScreenY	Integer	The offset of the FrameMaker window in pixels from the top of the screen (UNIX and Windows only). If you set a value that would result in the FrameMaker window being off the screen, that value is ignored and the old value is retained.
Snap	Integer	True if snap is turned on. <i>FM 6.0 or greater</i>
StackWarningLevel	Integer	This determines how warnings are displayed when history-clearing operations occur. It corresponds to an option set in the Preferences dialog, and to the preference-file flag <code>hpWarning</code> . The allowable values are: WarnNever - Disables warning for history-clearing operations for the session. WarnOnce - Displays a warning when a particular history-clearing command is issued, but does not warn on subsequent uses of that command. WarnAlways - Displays warnings every time a history-clearing command is issued. <i>Frame 7.2 or greater</i>
StructAppAttrConfigFile	StringList	FM 10 or greater
SuppressXMLParserWarnings	Integer	True/False. FM 10 or greater

Table 215: Session Properties (Page 6 of 6)

Property Name	Data Type	Description
TextColor (Read-Only)	Object	Text color object(Color Object). FM 10 or greater .
TmpDir (Read-Only)	String	Pathname of temporary directory for internal FrameMaker processes; the FrameMaker directory (on the Macintosh); the directory specified by the DOS \$TEMP environment variable (on Windows); the directory specified by \$TMPDIR (on UNIX).
UndoFDKRecording	Integer	True if multiple the undo feature is on, False otherwise. This property overrides the default value specified in the maker.ini flag EnableUndoInFDK . Enabling undo has a performance impact as data is cached for undo purposes. Set this property to False to disable Undo recording or True to enable it again. <i>Frame 7.2 or greater</i>
UseFMConsoleForErrorLog	Integer	Undocumented property. FM 11 or greater
UserHomeDir (Read-Only)	String	Pathname of the user's home directory (UNIX only).
UserLogin (Read-Only)	String	User login name.
UserName (Read-Only)	String	User name.
UserSettingsDir	String	Undocumented property.
VersionDescription	Integer	FM 2015 or greater .
VersionMajor (Read-Only)	Integer	Frame version number (before the decimal).
VersionMinor (Read-Only)	Integer	Frame version number (after the decimal).
ViewHotspotIndicators	Integer	Undocumented property. FM 11 or greater
ViewQuickAccessBar	Integer	True if the QuickAccess bar is visible.
ViewFormattingBar	Integer	True if the formatting bar is visible. ViewFormattingBar is available only on Windows platforms.
VisualCaretMovement	Integer	True/False FM 2015 or greater .
WindowSystem (Read-Only)	String	Name of window system that FrameMaker is running under: Macintosh MSWindows X Windows
XMLStructuredApplicationList (Read-Only)	StringList	List of Xml Structured applications. FM 10 or greater
XSLTProcessors (Read-Only)	StringList	Undocumented property. FM 11 or greater
XSLTTransformationScenarioFile	String	Undocumented property. FM 11 or greater

KeyCatalog

Object name - KeyCatalog (FM 11 or greater)

Table 216: KeyCatalog Properties

Property Name	Data Type	Description
IsDefault (Read-Only)	Integer	
IsStale	Integer	
KeyCatalogClientName	String	
KeyCount (Read-Only)	Integer	
NextKeyCatalogInSession (Read-Only)	Object	
NotLoaded	Integer	
Source	String	
SourceType	Integer	

Structured Objects

Object name - Element

Element objects have the following properties.

Table 217: Element Properties (Page 1 of 6)

Property Name	Data Type	Description
AllowAsSpecialCase	Integer	True if the element is treated as a special case.
AttrDisplay	Integer	Specifies element's attribute display properties: AttrDispNone: don't display attributes AttrDispReqspect: display required and specified attributes AttrDispAll: display all attributes
Attributes	AttributeList	The element's attributes.
AttributesEx	AttributeExList	The element's attributes(with original values and override info). FM 10 or greater
AttributeValueInvalid	Integer	True if the element contains an attribute value that is invalid.

Table 217: Element Properties (Page 2 of 6)

Property Name	Data Type	Description
BookComponent (Read-Only)	Object	Component file in book (BookComponent Object).
BookComponentMissing (Read-Only)	Integer	True if a component file is missing from a book.
ContentIsLooselyValid (Read-Only)	Integer	True if the content is loosely valid (it has some missing elements).
ContentIsStrictlyValid (Read-Only)	Integer	True if the content of the element is strictly valid.
ContentMustBeEmpty (Read-Only)	Integer	True if the element can't have any content.
ContentNeededAtBegin (Read-Only)	Integer	True if content is needed at the beginning of the element.
ContextLabel (Read-Only)	String	The context label (if any) applied to the element.
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
ElementDef	Object	Element's element definition (ElementDef Object).
ElementIsCollapsed	Integer	True if element is collapsed in Structure View.

Table 217: Element Properties (Page 3 of 6)

Property Name	Data Type	Description
ElementIsExcludedInContext (Read-Only)	Integer	True if the element is excluded.
ElementIsInvalidInParent (Read-Only)	Integer	True if the element cannot occur anywhere in its current parent.
ElementIsInvalidInPosition (Read-Only)	Integer	True if the element is invalid in its current position.
ElementIsUndefined (Read-Only)	Integer	True if the element is undefined.
ElementType (Read-Only)	Integer	The type of element: FoContainer FoTbl FoMarker FoEqn FoMmlEqn FM 12 or greater FoXRef FoTblTitle FoTblHeading FoTblBody FoTblFootnote FoTblRow FoTblCell FoFootnote FoGraphic FoSysvar FoRubiGroup FoRubi
ErrorInBookComponent (Read-Only)	Integer	True if there is a validation error for a component in a book.
FirstChildElement (Read-Only)	Object	If element is a container, element's first child element (Element Object).
FormatOverride†	Integer	True if the element has a format override.
FormattingLocked	Integer	True if the element has a format override.
HoleBeforeElement (Read-Only)	Integer	True if there are one or more missing elements before the element within the same parent.
InvalidHighestLevel (Read-Only)	Integer	True if the element cannot be the highest-level element in the flow.
LastChildElement (Read-Only)	Object	If element is a container, element's last child element (Element Object).
MatchesContextInUserString (Read-Only)	Integer	Compares the current context of the element's user string to the element's current context. If this user string specifies a CSS context that matches the element's current context, this property returns True .

Table 217: Element Properties (Page 4 of 6)

Property Name	Data Type	Description
MatchingFirstPgfClauses (Read-Only)	IntList	Objects of the first paragraph clauses (FmtRuleClause Objects) in the element's definition that apply to the element.
MatchingLastPgfClauses (Read-Only)	IntList	Object of the last paragraph clauses (FmtRuleClause Object) in the element's definition that apply to the element. ^a
MatchingObjectClauses (Read-Only)	IntList	Object of the object clauses (FmtRuleClause Object) in the element's definition that apply to the element.
MatchingPrefixClauses (Read-Only)	IntList	Object of the prefix clauses (FmtRuleClause Object) in the element's definition that apply to the element.
MatchingSuffixClauses (Read-Only)	IntList	Object of the suffix clauses (FmtRuleClause Object) in the element's definition that apply to the element.
MatchingTextClauses (Read-Only)	IntList	Object of the text clauses (FmtRuleClause Object) in the element's definition that apply to the element.
NameSpace	StringList	Prefix/path pairs defining namespaces for the element. This list must contain an even number of strings. For example: Prefix1, path1 Prefix2, path2 FM 7.0 or greater
NameSpaceScope (Read-Only)	Object	The Element which declares the namespace that is used to define the current element (Element Object). FM 7.0 or greater
NextDITAConrefElementInDoc (Read-Only)	Object	Undocumented. FM 11 or greater
NextDITALinkElementInDoc (Read-Only)	Object	Undocumented. FM 11 or greater
NextDITATopicrefElementInDoc (Read-Only)	Object	Undocumented. FM 11 or greater
NextDITATopicsetrefElementInDoc (Read-Only)	Object	Undocumented. FM 11 or greater
NextDITAXrefElementInDoc (Read-Only)	Object	Undocumented. FM 11 or greater
NextInvalidElement (Read-Only)	Object	Next invalid element in the document (Element Object).
NextSiblingElement (Read-Only)	Object	Element's next sibling element (Element Object).
NumNameSpaces (Read-Only)	Integer	The number of namespaces declared in the current element. FM 7.0 or greater

Table 217: Element Properties (Page 5 of 6)

Property Name	Data Type	Description
Object (Read-Only)	Object	Object of the object that an element contains. The type of object the Object specifies depends on the element definition as follows: FoTbl: Tbl FoMarker: Marker FoEqn: AFrame (containing the equation) FoXRef: XRef FoSysVar: Var FoFootnote: Fn FoGraphic: AFrame (containing the graphic) FoTblTitle: Tbl FoTblHeading: Tbl FoTblBody: Tbl FoTblFooting: Tbl FoTblRow: Row FoTblCell: Cell FoRubiGroup: Rubi FoRubi: Rubi
ParentElement (Read-Only)	Object	Element's parent element (Element Object).
PrevSiblingElement (Read-Only)	Object	Element's previous sibling element (Element Object).
TextRange (Read-Only)	TextRange	Text range that the element encompasses, see "Element Text Range Characteristics" on page 270.
TopicRefAbsolutePath (Read-Only)	String	The context label (if any) applied to the element.
Unique (Read-Only)	Integer	The element's UID.

Table 217: Element Properties (Page 6 of 6)

Property Name	Data Type	Description
UserString	String	A string to which clients can store private data.
ValidationFlags (Read-Only)	Integer	<p>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. For example, if the <code>ElementTypeMismatch</code> property is <code>True</code>, the <code>ElemTypeMismatch</code> flag is set.</p> <p> <code>ElemUndefined</code> <code>ElemTypeMismatch</code> <code>ElemExcluded</code> <code>ElemInvalidInParent</code> <code>ElemInvalidAtPosition</code> <code>ElemHasTextInvalid</code> <code>ElemContentMustBeEmpty</code> <code>ElemMissingContentBefore</code> <code>ElemMissingContentAtBeg</code> <code>ElemMissingContentAtEnd</code> <code>ElemNotValidAsRoot</code> <code>ElemBookCompMissing</code> <code>ElemBookCompInvalid</code> <code>ElemAttrValRequired</code> <code>ElemAttrValInvalid</code> <code>ElemContentStrictlyValid</code> <code>ElemContentLooselyValid</code> <code>ElemInvisibleAttrvalRequired</code> <code>ElemInvisibleAttrvalInvalid</code> </p> <p>Check these values using the bit operator <code>#&</code> as in:</p> <pre>IF elementVar.ValidationFlags #& ElemAttrValRequired DISPLAY 'This element is missing a required attribute'; ENDIF</pre>

- a. The `MatchingClauseTypeClauses` properties specify only format rule clauses that are in the element definition's format rules (that is, the format rules specified by the element definition's `TextFormatRules` and `ObjectFormatRules` properties). Format rule clauses that the element inherits from ancestor elements may also apply to it. To determine whether an element inherits format rule clauses from ancestor elements, you must traverse up the structure tree and check the `MatchingClauseTypeClauses` properties for each ancestor element.

The `TextRange` property for a structural element specifies an `TextRange` structure. The text locations specified by the `Begin` and `End` fields of the `TextRange` structure depend on the element type, as shown in the following table

Table 218: Element Text Range Characteristics

Element type	Text locations specified by the <code>Begin</code> and <code>End</code> fields
FOCONTAINER	Begin specifies the beginning of the container element, variable, or cross-reference; end specifies the end of the container element, variable, or cross-reference
FOSYSVAR	
FOXREF	
FOFOOTNOTE	Begin and End both specify the anchor of the object (the footnote, marker, table, graphic, or equation)
FOMARKER	
FOTBL	
FOGRAPHIC	
FOEQN	
FOTBLTITLE	
FOTBLHEADING	Begin and End both specify nothing (<code>TextRange</code> property fails)
FOTBLBODY	
FOTBLFOOTING	
FOTBLROW	
FOTBLCELL	
Text element	

Object name - `ElementDef`

`ElementDef` objects have the following properties

Table 219: Element Definition Properties (Page 1 of 3)

Property	Type	Meaning
<code>AlsoInsert</code>	<code>StringList</code>	The list of the tags of child elements that are automatically inserted when an element is initially added.
<code>AlsoInserts</code>	<code>StringList</code>	FM 10 or greater.
<code>AttributeDefs</code>	<code>AttributeDefList</code>	The element definition's attribute definitions.
<code>BannerText</code>	<code>String</code>	Undocumented. FM 11 or greater
<code>Comment</code>	<code>String</code>	Text string of comment.
<code>DescriptiveTag</code>	<code>String</code>	FM 10 or greater.

Table 219: Element Definition Properties (Page 2 of 3)

Property	Type	Meaning
ElementDefType	Integer	Type of formatter object represented by the element with element definition. CONTAINER identifies a container element. Other values identify object (noncontainer) elements. FOUNSPECIFIED or FOTEXTNODE (FM 10 or greater) FOCONTAINER FOTBL FOMARKER FOEQN FOXREF FOTBLTITLE FOTBLHEADING FOTBLBODY FOTBLFOOTING FOTBLROW FOTBLCELL FOFOOTNOTE FOGRAPHIC FOSYSVAR
ElementDescription	String	FM 10 or greater.
ElementInCatalog	Integer	True if the element is in the Element Catalog.
ElementPgFormat	String	The name of the paragraph format applied to the element.
Exclusions	StringList	List of excluded elements.
FirstPgRules†	IntList	The Objects of the first paragraph format rules (FmtRule Objects). ^a
GeneralRule	String	Text of the element's general rule.
GeneralRuleErrorOffsets (Read-Only)	IntList	Contains the error offsets (two positions are specified only if the content rule is ambiguous).
Inclusions	StringList	List of included elements.
InitStructurePattern	String	The initial structure pattern; for table elements, a comma delimited string that specifies the necessary child elements to automatically insert.
LastPgRules (Read-Only)	IntList	The Objects of the last paragraph format rules (FmtRule Objects).
Name†	String	Name of the element definition.
NextElementDefInDoc (Read-Only)	Object	Next element definition in the document's list of element definitions (ElementDef Object).
ObjectFmtRules (Read-Only)	IntList	The Objects of the object format rules (FmtRule Objects).
ObjectType	Integer	ObjectType is obsolete and no longer supported. Use ElementDefType instead.
PrefixRules (Read-Only)	IntList	The Objects of the prefix format rules (FmtRule Objects).

Table 219: Element Definition Properties (Page 3 of 3)

Property	Type	Meaning
SuffixRules (Read-Only)	IntList	The Objects of the suffix format rules (FmtRule Objects).
TableTagging	String	If the element is a table, the table format for new instances of the element.
TextFmtRules (Read-Only)	IntList	The Objects of the text format rules (FmtRule Objects).
ValidHighestLevel	Integer	True if the element can be used as the highest-level element for a flow.

a. To set the format rules for an element definition, use the New FmtRule. command

Object name - FmtChangeList

FmtChangeList objects have the following properties. Unlike other objects, FmtChangeList objects do not all have the same properties. All FmtChangeList objects have the properties listed here. However, each FmtChangeList object can have a different combination of the properties listed in the following sections as well as other text and paragraph properties.

Table 220: FmtChangeList Properties

Property Name	Data Type	Description
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
FmtChangeListInCatalog	Integer	True if the format change list is in the Format Change List Catalog. False if it is in an element definition, as part of the text format rules.
Name	String	The name of the format change list if it is in the Format Change List Catalog.
NextFmtChangeListInDoc	Object	The ID of the next format change list in the document (FmtChangeList ID).
PgfCatalogReference	String	A paragraph format tag if the format change list specifies one. If this property is set, you can't change any of the other format change list properties, except Name.

Object name - FmtRule

FmtRule objects have the following properties. Frame uses FmtRule objects to represent format rules in a structured FrameMaker document. It uses an FmtRuleClause object to represent each rule clause in a format rule.

Table 221: FmtRule Properties

Property Name	Data Type	Description
CountElements	StringList	If the format rule is a level rule, the list of element tags to count among the element's ancestors; the tags are specified by the <i>Count ancestors named</i> element of the format rule.
ElementDef	Object	If the format rule is not nested, the Object of the element definition that contains it (ElementDef Object).
FmtRuleClause	Object	If the format rule is nested, the Object of the format rule clause that contains it (FmtRuleClause Object).
FmtRuleClauses†	IntList	Objects of the format rule's format rule clause objects (FmtRuleClause Objects).
FmtRuleType	Integer	The format rule's type: ContextRule LevelRule
StopCountingAt	String	If the format rule is a level rule, the tag of the element at which to stop counting elements (the tag specified by the <i>Stop counting at first ancestor named</i> element).

Object name - FmtRuleClause

FmtRuleClause objects have the following properties.

Table 222: FmtRuleClause Properties

Property Name	Data Type	Description
ContextForCSS	IntList	Undocumented IntList option.
ContextLabel	String	The context label for generated files. It cannot contain white-space characters or any of these special characters: () & , * + ? < > % [] = ! ; : { } " When a user displays the Set Up dialog box to set up a generated file, the label appears next to elements to which the rule clause applies.
ElemPrefixSuffix	String	The text of the prefix or suffix. ElemPrefixSuffix is empty, if there is no prefix or suffix.
FmtChangeList†	Object	If the format rule clause specifies a format change list (RuleClauseType specifies RCChangeList), format change list Object (FmtChangeList Object). To change the FmtChangeList property, use New FmtRule command.
FmtChangeListTag	String	If the format rule clause specifies a change list (RuleClauseType specifies RCChangeListTag), the change list's tag.
FmtRule†	Object	The format rule Object containing the format rule clause (FmtRule Object).
FormatTag	String	The format tag if the format rule clause specifies one (RuleClauseType specifies RCTag). If IsTextRange is True, FormatTag specifies a character format tag; otherwise it specifies a paragraph tag, table tag, marker type, cross-reference format, or equation size.
IsTextRange	Integer	True if the container element is formatted as a text range instead of a paragraph.
RuleClauseType†	Integer	The type of rule clause: RCTag RCChangeListTag RCChangeList RCSubFmtRule
Specification	String	The format clause's context or level specification.
SpecificationForCSS	StringList	A List of CSS specifications that match the Specification property for the current clause. For example, assume an element has specification of F < (G H). This property returns the two following strings: G > F > E and H > F > E
SubFmtRule†	Object	If the format rule clause contains a nested format rule (RuleClauseType specifies RCSubFmtRule), the format rule's Object (FmtRule Object). To change the SubFmtRule property, use New FmtRule command.

SubColumns

Information for a Sub Column is stored in a SubCol object. You may retrieve a Sub Column object by going through a list stored in the containing Text Frame object.

Object name - SubCol

Table 223: Sub Column Properties (Page 1 of 2)

Property Name	Data Type	Description
ContentHeight (Read-Only)	Metric	The distance between the top of the column and the baseline of the last line in the column.
FirstAFrame (Read-Only)	Object	First anchored frame in the column (AFrame).
FirstCell (Read-Only)	Object	First table cell in the column (Cell).
FirstFn (Read-Only)	Object	First footnote in the column (Fn).
FirstPgf (Read-Only)	Object	First paragraph in the column (Pgf).
FrameParent (Read-Only)	Object	Text frame object that contains the column (TextFrame).
Height (Read-Only)	Metric	Column height.
LastAFrame (Read-Only)	Object	Last anchored frame in the column (AFrame).
LastCell (Read-Only)	Object	Last table cell in the column (Cell).
LastFn (Read-Only)	Object	Last footnote in the column (Fn).
LastPgf (Read-Only)	Object	Last paragraph in the column (Pgf).
LocX (Read-Only)	Metric	Offset from left side of the text frame that contains the column.
LocY (Read-Only)	Metric	Offset from top of text frame that contains the column.
NextSubCol (Read-Only)	Object	Next column in the text frame that contains the column (SubCol).
Overflowed (Read-Only)	Integer	True if the text frame containing the column has Autoconnect turned off and text overflows the column.
ParentTextFrame (Read-Only)	Object	Text frame object that contains the column (TextFrame).
PrevSubCol (Read-Only)	Object	Previous column in the text frame that contains the column (SubCol).

Table 223: Sub Column Properties (Page 2 of 2)

Property Name	Data Type	Description
StyleTag	String	Undocumented. FM 11 or greater
Width (<i>Read-Only</i>)	Metric	Column width.

Table Objects

The `Cell`, `Row`, and `Tbl` objects are used to represent various aspects of FrameMaker tables.

Object name - Cell

The `Cell` object is used to represent each cell in a table. `Cell` objects have the following properties.

Table 224: Table Cell Properties (Page 1 of 3)

Property Name	Data Type	Description
CellAboveInCol (<i>Read-Only</i>)	Object	Cell above current cell (<code>Cell</code> Object).
CellAngle	Integer	Angle of rotation.
CellBelowInCol (<i>Read-Only</i>)	Object	Cell below current cell (<code>Cell</code> Object).
CellColNum (<i>Read-Only</i>)	Integer	Cell's column number.
CellColumn (<i>Read-Only</i>)	Object	The column of the cell (<code>Column</code> Object). FM 2015 or greater
CellDefaultBottomRuling (<i>Read-Only</i>)	Object	Cell's default bottom ruling (<code>RulingFmt</code> Object).
CellDefaultLeftRuling (<i>Read-Only</i>)	Object	Cell's default left ruling (<code>RulingFmt</code> Object).
CellDefaultRightRuling (<i>Read-Only</i>)	Object	Cell's default right ruling (<code>RulingFmt</code> Object).
CellDefaultTopRuling (<i>Read-Only</i>)	Object	Cell's default top ruling (<code>RulingFmt</code> Object).
CellIsShown (<i>Read-Only</i>)	Integer	True if the cell is conditional and is visible.
CellIsSelected (<i>Read-Only</i>)	Integer	True if the cell is selected.
CellIsStraddled (<i>Read-Only</i>)	Integer	If the cell is in a straddle but is not the first cell, <code>CellIsStraddled</code> specifies <code>True</code> . If the cell is the first cell in a straddle, or it is not in a straddle, <code>CellIsStraddled</code> specifies <code>False</code> .
CellNumColsStraddled (<i>Read-Only</i>)	Integer	If the cell is the first cell in a horizontal straddle, <code>CellNumColsStraddled</code> specifies the number of columns in the straddle. Otherwise, it specifies <code>1</code> .

Table 224: Table Cell Properties (Page 2 of 3)

Property Name	Data Type	Description
CellNumRowsStraddled (<i>Read-Only</i>)	Integer	If the cell is the first cell in a vertical straddle, <code>CellNumRowsStraddled</code> specifies the number of rows in the straddle. Otherwise, it specifies 1.
CellOverrideBottomRuling	Object	Cell's bottom ruling (<code>RulingFmt</code> Object). ZERO if there is no override.
CellOverrideFill	Integer	Fill pattern. ZERO if there is no override fill pattern.
CellOverrideLeftRuling	Object	Cell's left ruling (<code>RulingFmt</code> Object). ZERO if there is no override left ruling.
CellOverrideRightRuling	Object	Cell's right ruling (<code>RulingFmt</code> Object). ZERO if there is no override right ruling.
CellOverrideShading	Object	Spot color (<code>Color</code> Object). ZERO if there is no override shading.
CellOverrideTopRuling	Object	Cell's top ruling (<code>RulingFmt</code> Object). ZERO if there is no override top ruling.
CellRow (<i>Read-Only</i>)	Object	The Row containing the cell (<code>Row</code> Object).
CellTbl (<i>Read-Only</i>)	Object	The Table containing the cell (<code>Tbl</code> Object).
CellUseOverrideBRuling	Integer	True if the cell's bottom ruling (specified by <code>CellOverrideBottomRuling</code>) overrides the default ruling specified by the table format.
CellUseOverrideFill	Integer	True if the cell's fill pattern (specified by <code>CellOverrideFill</code>) overrides the default fill pattern specified by the table format.
CellUseOverrideLRuling	Integer	True if the cell's left ruling (specified by <code>CellOverrideLeftRuling</code>) overrides the ruling specified by the table format.
CellUseOverrideRRuling	Integer	True if the cell's right ruling (specified by <code>CellOverrideRightRuling</code>) overrides the ruling specified by the table format.
CellUseOverrideShading	Integer	True if the cell's shading (specified by <code>CellOverrideShading</code>) overrides the default shading specified by the table format.
CellUseOverrideTRuling	Integer	True if the cell's top ruling (specified by <code>CellOverrideTopRuling</code>) overrides the default top ruling specified by the table format.
ContentHeight (<i>Read-Only</i>)	Metric	The distance between the top of the cell and the baseline of the last line in the cell.
Element (<i>Read-Only</i>)	Object	If the table cell is in an FM+SGML document, then this is the element object that contains it (<code>Element</code> Object).
FirstPgf (<i>Read-Only</i>)	Object	First paragraph in the cell (<code>Pgf</code> Object).
InTextFrame (<i>Read-Only</i>)	Object	Text frame containing the cell (<code>TextFrame</code> Object).
InTextObj (<i>Read-Only</i>)	Object	Text object containing the cell (<code>SubCol</code> Object).
IsCellSelected (<i>Read-Only</i>)	Integer	True if the cell is selected.
LastPgf (<i>Read-Only</i>)	Object	The Last paragraph in the cell (<code>Pgf</code> Object).
NextCell	Object	The Next cell in the text frame (<code>Cell</code> Object).
NextCellInRow (<i>Read-Only</i>)	Object	The Next cell in current row from left to right (<code>Cell</code> Object). If the current cell is the last cell of the row, then this value is NULL

Table 224: Table Cell Properties (Page 3 of 3)

Property Name	Data Type	Description
NextCellInTbl (<i>Read-Only</i>)	Object	The Next cell from left to right (Cell Object). If the cell is at the end of a row, the next cell is the first cell in the next row. If the current cell is the last cell of the table, then this value is NULL.
NextSelectedCellInTbl (<i>Read-Only</i>)	Object	The Next selected cell from left to right (Cell Object). If the current cell is the last cell of the selection, this value will be NULL.
Overflowed (<i>Read-Only</i>)	Integer	Specifies whether the text in the cell overflows. True if the row Height Limit Maximum is too low to display all the text in the cell.
PrevCell (<i>Read-Only</i>)	Object	Previous cell in the text frame (Cell Object).
PrevCellInRow (<i>Read-Only</i>)	Object	Previous cell in current row (Cell Object). If the current cell is the first cell in the row, then this value is NULL.
PrevCellInTbl (<i>Read-Only</i>)	Object	The Previous cell from right to left (Cell Object). If the cell is at the beginning of a row, the previous cell is the last cell in the previous row. If the current cell is the first cell of the table, then this value is NULL.
PrevSelectedCellInTbl (<i>Read-Only</i>)	Object	The Previous selected cell from right to left (Cell Object). If the current cell is the first cell of the selection, this value will be NULL.
Text	String	The text in a table cell.

Object name - Column (FM 2015 or greater)

The Column object is used to represent each column in a table. Column objects have the following properties.

Table 225: Table Column Properties

Property Name	Data Type	Description
ColumnIsShown	Integer	True if the column is visible.
ColumnNum	Integer	Number of the column
ColumnTbl (<i>Read-Only</i>)	Object	Table of the column (Table Object).
NextColumnInTbl (<i>Read-Only</i>)	Object	Next column in the table (Column Object).
NextVisibleColumnInTbl (<i>Read-Only</i>)	Object	Next visible column in the table (Column Object).
PrevColumnInTbl (<i>Read-Only</i>)	Object	Previous column in the table (Column Object).
PrevVisibleColumnInTbl (<i>Read-Only</i>)	Object	Previous visible column in the table (Column Object).

Object name - Row

The Row object is used to represent each row in a table. Row objects have the following properties.

Table 226: Table Row Properties (Page 1 of 2)

Property Name	Data Type	Description
CondFmtIsShown	Integer	True if the condition is shown.
Element (Read-Only)	Object	If the table row is in an FM+SGML document, then this is the element object that contains it (Element Object).
FirstCellInRow (Read-Only)	Object	First cell in the row (Cell Object).
Height (Read-Only)	Metric	Height of the row.
InCond	IntList	List of condition codes applicable at the this text location. This is a list of integers. You must use the New Object command to convert one of these integers into a condition object.
InCond	IntList	Condition tags for row (array of CondFmt Objects).
LastCellInRow (Read-Only)	Object	Last cell in the row (Cell Object).
LocX (Read-Only)	Metric	Offset from the left side of the text frame containing the row.
LocY (Read-Only)	Metric	Offset from the top of the page frame containing the row.
NextRowInTbl (Read-Only)	Object	Next row (Row Object) in the table.
PrevRowInTbl (Read-Only)	Object	Previous row (Row Object) in the table.
RowHasSelection (Read-Only)	Integer	True if the row has cells that are part of the selection.
RowIsShown (Read-Only)	Integer	True if the conditional row is shown.
RowKeepWithNext	Integer	True if Keep With Next Row is enabled.
RowKeepWithPrev	Integer	True if Keep With Previous Row is enabled.
RowMaxHeight	Metric	Maximum row height.
RowMinHeight	Metric	Minimum row height.
RowStart	Integer	Row placement: RowAnywhere RowTopOfCol RowTopOfPage RowTopOfLeftPage RowTopOfRightPage

Table 226: Table Row Properties (Page 2 of 2)

Property Name	Data Type	Description
RowTbl (Read-Only)	Object	Table containing the row (Tbl Object).
RowType (Read-Only)	Integer	Type of row: RowHeading RowBody RowFooting
SepOverride	Object	Color separation format override (Color Object).
StyleOverrides	Integer	Style condition indicators for conditional text: CsDoubleUnderline CsNoOverride CsOverline CsSingleUnderline CsStrikeThrough All style condition indicators are represented as hatched lines for the table rows.
UseSepOverride	Integer	True if SepOverride property overrides default from the table.
Width (Read-Only)	Metric	Width of the row.

Object name - Tbl

The Tbl object represents a FrameMaker table. It has the following properties.

Table 227: Table Properties (Page 1 of 5)

Property Name	Data Type	Description
BottomRowSelection (Read-Only)	Object	Bottom body row in selection, if table is selected (Row Object)
ContentHeight (Read-Only)	Metric	The height of the table title.
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
Element (Read-Only)	Object	If the table is in an FM+SGML document, then this is the element object that os associated with it (Element Object).
FirstAFrame (Read-Only)	Object	First anchored frame in the table (AFrame Object)

Table 227: Table Properties (Page 2 of 5)

Property Name	Data Type	Description
FirstCellInTbl (Read-Only)	Object	The first cell in the table (Cell Object)
FirstColumnInSelection (Read-Only)	Object	The first Column in the table selection (Column Object) FM 2015 or greater
FirstColumnInTbl (Read-Only)	Object	The first Column in the table (Column Object) FM 2015 or greater
FirstFn (Read-Only)	Object	The first footnote in the table (Fn Object)
FirstPgf (Read-Only)	Object	If table has a title, the first paragraph in the title (Pgf Object).
FirstRowInTbl (Read-Only)	Object	The first row in the table (Row Object)
FirstSelectedCellInTbl (Read-Only)	Object	The first selected cell in the table (Cell Object). If no cell is selected, then this value is NULL.
FirstVisibleColumnInTbl (Read-Only)	Object	The first Visible Column in the table (Column Object) FM 2015 or greater
LastAFrame (Read-Only)	Object	The last anchored frame in the table (AFrame Object)
LastCellInTbl (Read-Only)	Object	The last cell in the table (Cell Object)
LastColumnInSelection (Read-Only)	Object	The last Column in the table selection (Column Object) FM 2015 or greater
LastColumnInTbl (Read-Only)	Object	The last Column in the table (Column Object) FM 2015 or greater
LastFn (Read-Only)	Object	The last footnote in the table (Fn Object)
LastPgf (Read-Only)	Object	If table has a title, the last paragraph in the title (Pgf Object).
LastRowInTbl (Read-Only)	Object	The last row in the table (Row Object)
LastSelectedCellInTbl (Read-Only)	Object	The last selected cell in the table (Cell Object). If no cell is selected, then this value is NULL.
LastVisibleColumnInTbl (Read-Only)	Object	The last Visible Column in the table (Column Object) FM 2015 or greater
LeftColNum (Read-Only)	Integer	Number of leftmost selected column, if table is selected (columns are numbered from left to right, starting with 0)
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.
NextTblInDoc (Read-Only)	Object	Next table (Tbl Object) in the document
OrphanRows	Integer	Number of orphan rows.
Overflowed (Read-Only)	Integer	True if the table has cells that are not shown because they extend beyond the text frame boundaries.

Table 227: Table Properties (Page 3 of 5)

Property Name	Data Type	Description
RightColNum (<i>Read-Only</i>)	Integer	Number of rightmost selected column, if table is selected (columns are numbered from left to right, starting with 0)
TblAlignment	Integer	Horizontal placement of table: AlignTblCenter AlignTblLeft AlignTblRight AlignTblSideCloserToBinding AlignTblSideFartherFromBinding
TblBodyElement (<i>Read-Only</i>)	Object	If the table is in an FM+SGML document, then this is the element object that contains the body rows (Element Object).
TblBodyFirstColor	Object	First spot color for table body (Color Object)
TblBodyFirstFill	Integer	First fill pattern for table body
TblBodyFirstPeriod	Integer	Number of columns or body rows to which the first fill pattern (specified by TblBodyFirstFill) is applied
TblBodyNextColor	Object	Exception color for columns or body rows (Color Object)
TblBodyNextFill	Integer	Exception fill pattern for table body
TblBodyNextPeriod	Integer	Number of columns or body rows to which the exception fill pattern (specified by TblBodyNextFill) is applied
TblBodyRowRuling	Object	Ruling applied to body rows specified by TblBodyRowRulingPeriod (RulingFmt Object).
TblBodyRowRulingPeriod	Integer	The periodicity of the ruling specified by TblBodyRowRuling . For example, if TblBodyRowRulingPeriod is set to 3, the ruling specified by TblBodyRowRuling is applied to every third row.
TblBodyShadeBy	Integer	True if Shade By is set to Columns ; False if Shade By is set to Rows
TblBottomRuling	Integer	Ruling applied to the bottom of the table (RulingFmt Object).
TblCatalogEntry (<i>Read-Only</i>)	Integer	True if the table's format is in the Table Catalog
TblCellBottomMargin	Metric	Default bottom cell margin for the table
TblCellLeftMargin	Metric	Default left cell margin for the table
TblCellRightMargin	Metric	Default right cell margin for the table
TblCellTopMargin	Metric	Default top cell margin for the table
TblColRuling	Object	Ruling applied to table columns specified by TblColRulingPeriod (RulingFmt Object).
TblColRulingPeriod	Integer	The periodicity of the ruling specified by TblColRuling . For example, if TblColRulingPeriod is set to 2, the ruling specified by TblColRuling is applied to every other column.
TblColWidths	MetricList	List of column widths
TblElement (<i>Read-Only</i>)	Object	If the table is in an FM+SGML document, then this is the element object that contains the table (Element Object).
TblFooterElement (<i>Read-Only</i>)	Object	If the table is in an FM+SGML document, then this is the element object that contains the header rows (Element Object).

Table 227: Table Properties (Page 4 of 5)

Property Name	Data Type	Description
TblHeaderElement (Read-Only)	Object	If the table is in an FM+SGML document, then this is the element object that contains the footer rows (Element Object).
TblHFColor	Object	Color for table heading and footing
TblHFFill	Integer	Fill pattern for table heading and footing (integer percentage)
TblHFRowRuling	Object	Ruling for table heading and footing rows (RulingFmt Object).
TblHFSeparatorRuling	Object	Separator ruling for table heading and footing rows (RulingFmt Object).
TblLastBodyRuling	Integer	True if Draw Bottom Ruling on Last Sheet Only is enabled (RulingFmt Object).
TblLeftIndent	Metric	Left indent for the table
TblLeftRuling	Object	Ruling for the left side of the table (RulingFmt Object).
TblNumbering	Integer	Direction of autonumbering for the table: TblNumByCol TblNumByRow
TblNumCols (Read-Only)	Integer	Number of columns in the table
TblNumRows (Read-Only)	Integer	Number of rows in the table
TblOtherBodyRowRuling	Object	Ruling for body rows that aren't specified by TblBodyRowRulingPeriod (RulingFmt Object).
TblOtherColRuling	Object	Ruling for columns that aren't specified by TblColRulingPeriod (RulingFmt Object).
TblPlacement	Integer	Vertical placement of table on page: TblAnywhere TblTopOfCol TblTopOfPage TblTopOfLeftPage TblTopOfRightPage TblFloat
TblRightIndent	Metric	Right indent for the table
TblRightRuling	Object	Ruling for the right side of the table (RulingFmt Object).
TblSpaceAbove	Metric	Vertical space above the table
TblSpaceBelow	Metric	Vertical space below the table
TblTag	String	Name of table format
TblTitleElement (Read-Only)	Object	If the table is in an FM+SGML document, then this is the element object that contains the table title (Element Object).
TblTitleGap	Metric	Gap between the title and top or bottom row of the table.
TblTitlePosition	Integer	The placement of the table title: TblNoTitle: table has no title TblTitleBelow: the title appears below the table TblTitleAbove: the title appears above the table

Table 227: Table Properties (Page 5 of 5)

Property Name	Data Type	Description
TblTitleSelected (Read-Only)	Integer	True if table title is selected
TblTitleSelected (Read-Only)	Integer	True if the table title is selected.
TblTopRuling	Object	Ruling for the top of the table (RulingFmt Object).
TblWidth (Read-Only)	Metric	Horizontal width of the table
TextLoc	TextLoc	The text location of the table's anchor
TopRowSelection (Read-Only)	Object	Top row in selection, if table is selected (Row Object)

Object name - TblFmt

The TblFmt object is used to represent each table format in a document. TblFmt objects have the following properties.

Table 228: Table Format Properties (Page 1 of 3)

Property Name	Data Type	Description
Direction	Integer	FM 2015 or greater DirInherit DirLtr DirRtl.
ResolvedDirection (Read-Only)	Integer	FM 2015 or greater DirLtr DirRtl.
Name	String	Name of the paragraph format of the table title
NextTblFmtInDoc (Read-Only)	Object	Next table format in the document (TblFmt Object)
OrphanRows	Integer	Number of orphan rows
TblAlignment	Integer	Horizontal placement of table: AlignTblCenter AlignTblLeft AlignTblRight
TblBodyFirstColor	Object	First spot color for table body (Color Object)
TblBodyFirstFill	Integer	First fill pattern for table body
TblBodyFirstPeriod	Integer	Number of columns or body rows to which the first fill pattern (specified by TblBodyFirstFill) is applied
TblBodyNextColor	Object	Exception color for columns or body rows (Color Object)
TblBodyNextFill	Integer	Exception fill pattern for table body

Table 228: Table Format Properties (Page 2 of 3)

Property Name	Data Type	Description
TblBodyNextPeriod	Integer	Number of columns or body rows to which the exception fill pattern (specified by TblBodyNextFill) is applied
TblBodyRowRuling	Object	Ruling for body rows that aren't specified by TblBodyRowRulingPeriod (RulingFmt Object).
TblBodyRowRulingPeriod	Integer	The periodicity of the ruling specified by TblOtherBodyRowRuling . For example, if TblBodyRowRulingPeriod is set to 3, the ruling specified by TblOtherBodyRowRuling is applied to every third row.
TblBodyShadeBy	Integer	True if Shade By is set to Columns; False if Shade By is set to Rows
TblBottomRuling	Object	Ruling for the bottom of the table (RulingFmt Object).
TblCatalogEntry	Integer	True if format is in the Table Catalog
TblCellBottomMargin	Metric	Default bottom cell margin for the table.
TblCellLeftMargin	Metric	Default left cell margin for the table.
TblCellRightMargin	Metric	Default right cell margin for the table.
TblCellTopMargin	Metric	Default top cell margin for the table.
TblColRuling	Object	Ruling for columns that aren't specified by TblColRulingPeriod (RulingFmt Object).
TblColRulingPeriod	Integer	The periodicity of the ruling specified by TblOtherColRuling . For example, if TblColRulingPeriod is set to 2, the ruling specified by TblOtherColRuling is applied to every other column.
TblHFColor	Object	Color for table heading and footing
TblHFFill	Integer	Fill pattern for table heading and footing (integer percentage)
TblHFRowRuling	Object	Ruling for the heading and footing rows (RulingFmt Object).
TblHFSeparatorRuling	Object	Separator ruling for the table heading and footing rows (RulingFmt Object).
TblInitNumBodyRows	Integer	Number of body rows for new table
TblInitNumCols	Integer	Number of columns for new table
TblInitNumFRows	Integer	Number of footing rows for new table
TblInitNumHRows	Integer	Number of heading rows for new table
TblLastBodyRuling	Integer	True if Draw Bottom Ruling on Last Sheet Only is enabled (RulingFmt Object).
TblLeftIndent	Metric	Left indent for the table
TblLeftIndent	Metric	Left indent for table
TblLeftRuling	Object	Ruling for the left side of the table (RulingFmt Object).
TblNumbering	Integer	Direction of autonumbering for the table: TblNumByCol TblNumByRow
TblOtherBodyRowRuling	Object	Ruling applied to body rows specified by TblBodyRowRulingPeriod (RulingFmt Object).
TblOtherColRuling	Object	Ruling applied to table columns specified by TblColRulingPeriod (RulingFmt Object).

Table 228: Table Format Properties (Page 3 of 3)

Property Name	Data Type	Description
TblPlacement	Integer	Vertical placement of table on page: TblAnywhere TblTopOfCol TblTopOfPage TblTopOfLeftPage TblTopOfRightPage TblFloat
TblRightIndent	Metric	Right indent for table
TblRightRuling	Object	Ruling for the right side of the table (RulingFmt Object).
TblSpaceAbove	Metric	Vertical space above table
TblSpaceBelow	Metric	Vertical space below table
TblTag	String	Name of the table format
TblTitleGap	Metric	Gap between title and top or bottom row
TblTitlePosition	Integer	Placement of table title: TblNoTitle : table has no title TblTitleBelow : the title appears below the table TblTitleAbove : the title appears above the table
TblTopRuling	Object	Ruling for the top of the table (RulingFmt Object).

Table ruling formats

The `RulingFmt` object is used to represent each ruling format in a document. Tables and table formats specify rulings by specifying the `RulingFmt` objects.

Object name - RulingFmt

`RulingFmt` objects have the following properties

Table 229: Ruling Format Properties (Page 1 of 2)

Property Name	Data Type	Description
Name	String	Ruling format name.
NextRulingFmtInDoc (Read-Only)	Object	Next ruling format in document (RulingFmt Object).
Pen	Integer	The pen pattern (numbers between 0 and 7). Pen patterns: FillBlack FillWhite FillClear

Table 229: Ruling Format Properties (Page 2 of 2)

Property Name	Data Type	Description
RulingGap	Metric	Gap between double ruling lines (0.015 pt to 360 pt).
RulingLines	Integer	Number of ruling lines (0 to 2 lines).
RulingPenWidth	Metric	Ruling line thickness (0.015 pt to 360 pt).
RulingSep	Object	Spot color of ruling format (Color Object).

Text Insets

The `TiApiClient`, `TiFlow`, `TiText`, and `TiTextTable` objects are used to represent text that is imported by reference (text insets).

Object name - TiApiClient

An `TiApiClient` object represents text imported by a client. `TiApiClient` objects have the following properties

Table 230: TiApiClient Properties (Page 1 of 2)

Property Name	Data Type	Description
LastUpdate	Integer	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
Name	String	A name assigned to the inset by a client. It is not automatically assigned by FrameMaker.
NextTiInDoc (<i>Read-Only</i>)	Object	The next text inset object in the list of text insets in the document (<code>TiApiClient</code> , <code>TiText</code> , <code>TiTextTable</code> , or <code>TiFlow</code> Object).
TextRange (<i>Read-Only</i>)	TextRange	The text range, in the document containing the text inset, occupied by the text inset.
TiAutomaticUpdate	Integer	True if the inset is updated automatically. <code>TiAutomaticUpdate</code> has no effect if the document's <code>DontUpdateTextInsets</code> property is set to True.
TiClientData	String	Data used by the client (for example, an SQL query).
TiClientName	String	The registered name of the client that created the inset.
TiClientSource	String	The name that appears as the source in the Text Inset Properties dialog box.
TiClientType	String	The name that appears as the source type in the Text Inset Properties dialog box.
TiClientUpdateSilent	Integer	Undocumented. FM 11 or greater
TiFile	String	Pathname of the source file.
TiFileModDate	String	The modification date of the text inset's source file.
TiIsNested (<i>Read-Only</i>)	Integer	True if the inset is nested inside another inset and False otherwise.
TiIsUnresolved	Integer	True if the inset is unresolved. A client should set this property to True if it is unable to resolve the inset.

Table 230: TiApiClient Properties (Page 2 of 2)

Property Name	Data Type	Description
TiLocked	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.
TiMacEdition (Read-Only)	Integer	If the source file is a Macintosh edition, the ID of its sect and alis records.
Unique (Read-Only)	Integer	The Inset's UID.

Object name - TiFlow

An TiFlow object represents text imported from a FrameMaker document or a MIF file. TiFlow objects have the following properties

Table 231: TiFlow Properties (Page 1 of 2)

Property Name	Data Type	Description
LastUpdate	Integer	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
Name	String	A name assigned to the inset by a client. It is not automatically assigned by FrameMaker.
NextTiInDoc (Read-Only)	Object	The next text inset object in the list of text insets in the document (TiApiClient, TiText, TiTextTable, or TiFlow Object).
TextRange (Read-Only)	TextRange	The text range, in the document containing the text inset, occupied by the text inset.
TiAutomaticUpdate	Integer	True if the inset is updated automatically. TiAutomaticUpdate has no effect if the document's DontUpdateTextInsets property is set to True.
TiFile	String	Pathname of the source file.
TiFileModDate	String	The modification date of the text inset's source file.
TiFlowName	String	The name of the imported flow if TiMainFlow is False. If the source file is an edition, TiFlowName is set to Macintosh edition.
TiFlowPageSpace	Integer	The type of pages the imported flow is on: BodyPage ReferencePage
TiFormat	Integer	Source of the imported text's format: SourceDoc: the text is formatted with formats from the source document PlainText: the text is formatted as plain text EnclosingDoc: the text is formatted with formats from the document into which it is imported
TiIsNested (Read-Only)	Integer	True if the inset is nested inside another inset and False otherwise.
TiIsUnresolved	Integer	True if the inset is unresolved. A client should set this property to True if it is unable to resolve the inset.
TiLocked	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.

Table 231: TiFlow Properties (Page 2 of 2)

Property Name	Data Type	Description
TiMacEdition (<i>Read-Only</i>)	Integer	If the source file is a Macintosh edition, the ID of its <code>sect</code> and <code>alis</code> records.
TiMainFlow	Integer	<code>True</code> if the inset text is imported from the main flow of the source document.
TiRemoveOverrides	Integer	<code>True</code> if format overrides are removed from the text when <code>TiFormat</code> is set to <code>EnclosingDoc</code> .
TiRemovePageBreaks	Integer	<code>True</code> if page breaks are removed from the text when <code>TiFormat</code> is set to <code>EnclosingDoc</code> .
Unique (<i>Read-Only</i>)	Integer	The Inset's UID.

Object name - TiText properties

An `TiText` object represents text imported from a text file. `TiText` objects have the following properties.

Table 232: TiText Properties

Property Name	Data Type	Description
LastUpdate	Integer	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
Name	String	A name assigned to the inset by a client. It is not automatically assigned by FrameMaker.
NextTiInDoc (<i>Read-Only</i>)	Object	The next text inset object in the list of text insets in the document (<code>TiApiClient</code> , <code>TiText</code> , <code>TiTextTable</code> , or <code>TiFlow</code> Object).
TextRange (<i>Read-Only</i>)	TextRange	The text range, in the document containing the text inset, occupied by the text inset.
TiAutomaticUpdate	Integer	<code>True</code> if the inset is updated automatically. <code>TiAutomaticUpdate</code> has no effect if the document's <code>DontUpdateTextInsets</code> property is set to <code>True</code> .
TiEOLisEOP	Integer	<code>True</code> if line ends in the imported text file are treated as paragraph ends.
TiFile	String	Pathname of the source file.
TiFileModDate	String	The modification date of the text inset's source file.
TiIsNested (<i>Read-Only</i>)	Integer	<code>True</code> if the inset is nested inside another inset and <code>False</code> otherwise.
TiIsUnresolved	Integer	<code>True</code> if the inset is unresolved. A client should set this property to <code>True</code> if it is unable to resolve the inset.
TiLocked	Integer	<code>True</code> 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.
TiMacEdition (<i>Read-Only</i>)	Integer	If the source file is a Macintosh edition, the ID of its <code>sect</code> and <code>alis</code> records.
TiTextEncoding (<i>Read-Only</i>)	String	The <code>ImportHintString</code> for the text inset. If this is not a <code>TiText</code> or <code>TiTextTable</code> , the string is null.
Unique (<i>Read-Only</i>)	Integer	The Inset's UID.

Object name - TiTextTable properties

An `TiTextTable` object represents text imported from a text file into a table. `TiTextTable` objects have the following properties

Table 233: TiTextTable Properties

Property Name	Data Type	Description
<code>LastUpdate</code>	<code>Integer</code>	Time when the inset was last updated, expressed in seconds since 1 January, 1970.
<code>Name</code>	<code>String</code>	A name assigned to the inset by a client. It is not automatically assigned by FrameMaker.
<code>NextTiInDoc</code> (<i>Read-Only</i>)	<code>Object</code>	The next text inset object in the list of text insets in the document (<code>TiApiClient</code> , <code>TiText</code> , <code>TiTextTable</code> , or <code>TiFlow</code> Object).
<code>TextRange</code> (<i>Read-Only</i>)	<code>TextRange</code>	The text range, in the document containing the text inset, occupied by the text inset.
<code>TiAutomaticUpdate</code>	<code>Integer</code>	<code>True</code> if the inset is updated automatically. <code>TiAutomaticUpdate</code> has no effect if the document's <code>DontUpdateTextInsets</code> property is set to <code>True</code> .
<code>TiByRows</code>	<code>Integer</code>	<code>True</code> if each paragraph in the imported text is converted to a row of table cells; <code>False</code> if each paragraph in the imported text is converted to a table cell.
<code>TiFile</code>	<code>String</code>	Pathname of the source file.
<code>TiFileModDate</code>	<code>String</code>	The modification date of the text inset's source file.
<code>TiHeadersEmpty</code>	<code>Integer</code>	<code>True</code> if the imported text is not used to fill the heading rows.
<code>TiIsNested</code> (<i>Read-Only</i>)	<code>Integer</code>	<code>True</code> if the inset is nested inside another inset and <code>False</code> otherwise.
<code>TiIsUnresolved</code>	<code>Integer</code>	<code>True</code> if the inset is unresolved. A client should set this property to <code>True</code> if it is unable to resolve the inset.
<code>TiLocked</code>	<code>Integer</code>	<code>True</code> 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>TiMacEdition</code> (<i>Read-Only</i>)	<code>Integer</code>	If the source file is a Macintosh edition, the ID of its <code>sect</code> and <code>alis</code> records.
<code>TiNumCols</code>	<code>Integer</code>	If <code>TiByRows</code> is <code>False</code> , the number of columns in the table.
<code>TiNumHeaderRows</code>	<code>Integer</code>	The number of heading rows in the table.
<code>TiNumSeparators</code>	<code>Integer</code>	If <code>TiSeparator</code> specifies a space, the number of spaces used as a separator to parse the text into table cells.
<code>TiSeparator</code>	<code>String</code>	If <code>TiByRows</code> is <code>True</code> , a string specifying a separator, such as a tab, used to parse the text into table cells.
<code>TiTblTag</code>	<code>String</code>	The table format tag of the imported table.
<code>TiTextEncoding</code> (<i>Read-Only</i>)	<code>String</code>	The <code>ImportHintString</code> for the text inset. If this is not a <code>TiText</code> or <code>TiTextTable</code> , the string is null.
<code>Unique</code> (<i>Read-Only</i>)	<code>Integer</code>	The Inset's UID.

Text properties

Text has the following properties. To retrieve these properties for a text location, use the `Get TextProperties` command. The properties these functions return apply to the character to the right of the location you specify. Use the `Apply TextProperties` command to set the text properties over a range of text.

Table 234: Text Properties (Page 1 of 3)

Property Name	Data Type	Description
BkColor	Object	Background color object(<code>Color</code> Object). FM 10 or greater .
Capitalization	Integer	Type of capitalization: CapitalCaseNorm CapitalCaseSmall CapitalCaseLower CapitalCaseUpper
ChangeBar	Integer	True if Change Bars is enabled at the text location.
CharTag	String	Name of character format tag applied to text location.
Color	Object	Spot color (<code>Color</code> Object).
CombinedFontFamily	Object	Combined font definition (<code>CombinedFontDefn</code>)
FontAngle	Integer	Font angle (specifies an index into the array of font angles provided by the session property <code>FontAngleNames</code>).
FontFamily	Integer	Font family (specifies an index into the array of font families provided by the session property <code>FontFamilyNames</code>).
FontPlatformName	String	Name that uniquely identifies a font on a specific platform.
FontPostScriptName	String	Name given to a font when it is sent to a PostScript printer.
FontSize	Metric	Font size (2 pt to 400 pt).
FontVariation	Integer	Font variation (specifies an index into the array of font variations provided by the session property <code>FontVariationNames</code>).
FontWeight	Integer	Font weight (specifies an index into the array of font weights provided by the session property <code>FontWeightNames</code>).
Height (Read-Only)	Metric	Height of text at text location.
InCond	IntList	List of condition codes applicable at the this text location. This is a list of integers. You must use the <code>New Object</code> command to convert one of these integers into a condition object.
InCond	IntList	Condition tags that apply to the text (array of <code>CondFmt</code> Objects).
InTextFrame (Read-Only)	Object	Text frame containing the text (<code>TextFrame</code> Object).
InTextObj (Read-Only)	Object	Text frame or text line in which text appears (<code>TextFrame</code> or <code>TextLine</code> Object).
KernX	Metric	Horizontal kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves a character right, and a negative value moves a character left.
KernY	Metric	Vertical kern value for manual kerning expressed as a percentage of an em (metric –100% to 1000%). A positive value moves characters up, and a negative value moves characters down.

Table 234: Text Properties (Page 2 of 3)

Property Name	Data Type	Description
LineAscent (Read-Only)	Metric	The ascent of the line containing the text, measured from the line's baseline.
LineBaseLine (Read-Only)	Metric	The location of the line containing the text, measured from the top of the object containing the text.
LineDescent (Read-Only)	Metric	The descent of the line containing the text, measured from the line's baseline.
Locked	Integer	True if the text is included in a text inset that gets its formatting from the source document.
LocX (Read-Only)	Metric	Offset of the left side of a character from the left side of the subcolumn (SubCol object) containing it.
LocY (Read-Only)	Metric	Offset of the top of a character from the top of the subcolumn (SubCol object) containing it.
Outline	Integer	True if Outline property is enabled (Macintosh only).
Overline	Integer	True if Overline is enabled.
PairKern	Integer	True if Pair Kern is enabled.
Position	Integer	Position of the text: PosSuper: Superscript PosNorm: Baseline PosSub: Subscript
ReviewerName	String	Name of the reviewer. FM 10 or greater
ReviewTime	String	Time of the review. FM 10 or greater
SepOverride	Object	Color for separation override (Color Object).
Shadow	Integer	True if Shadow is enabled (Macintosh only).
Spread	Metric	FrameMaker 5.1x only. FrameMaker 5.5 use the "tracking" property.
Stretch	Metric	Character stretch (set width) expressed as a percentage of normal stretch for the font (metric -10% to 1000%).
Strikethrough	Integer	True if Strikethrough is enabled.
StyleOverrides	Integer	Bit flags that specify which overrides (condition indicators) are used: CsChangebar CsNoOverride CsOverline CsStrikeThrough CsSingleUnderline CsDoubleUnderline
Underlining	Integer	Type of underlining: CbNoUnderline CbSingleUnderline CbDoubleUnderline CbNumericUnderline
UseBkColor	Integer	True/False. FM 10 or greater

Table 234: Text Properties (Page 3 of 3)

Property Name	Data Type	Description
UseSepOverride	Integer	True if color specified for <code>SepOverride</code> overrides default.
Width (<i>Read-Only</i>)	Metric	Width of a character

Variables

The `Var` object is used to represent a variable instance and a `VarFmt` object is used to represent a variable format.

Object name - Var

`Var` objects have the following properties.

Table 235: Variable Properties

Property Name	Data Type	Description
Element (<i>Read-Only</i>)	Object	If the table is in an FM+SGML document, then this is the element object that contains the Variable (<code>Element</code> Object).
Locked	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.
NextVarInDoc (<i>Read-Only</i>)	Object	Next variable instance (<code>Var</code> Object) in the document.
TextRange (<i>Read-Only</i>)	TextRange	The text range the variable instance encompasses.
VarFmt	Object	The variable instance's format (<code>VarFmt</code> Object).

Object name - VarFmt

`VarFmt` objects have the following properties.

Table 236: Variable Format Properties (Page 1 of 2)

Property Name	Data Type	Description
Fmt	String	The variable format definition; the building blocks and text strings used to create a variable instance with the variable format.
Name	String	The variable format's name.

Table 236: Variable Format Properties (Page 2 of 2)

Property Name	Data Type	Description
NextVarFmtInDoc (<i>Read-Only</i>)	Object	Next variable format (VarFmt Object) in the document's list of variable formats.
SystemVar (<i>Read-Only</i>)	Integer	<p>The variable format's type.</p> <p>VarUserVariable: a user-defined variable format.</p> <p>The following types specify system variable formats:</p> <p>VarCurrentPageNum</p> <p>VarPageCount</p> <p>VarChapterNum</p> <p>VarVolumeNum</p> <p>VarCurrentDateLong</p> <p>VarCurrentDateShort</p> <p>VarModificationDateLong</p> <p>VarModificationDateShort</p> <p>VarCreationDateLong</p> <p>VarCreationDateShort</p> <p>VarFileNameLong</p> <p>VarFileNameShort</p> <p>VarHeaderFooterN</p> <p style="padding-left: 40px;">where N is from 1 to 4</p> <p style="padding-left: 40px;">or where N is from 5 to 12 (FM 7.0 or greater)</p> <p>VarTableContinuation</p> <p>VarTableSheet</p>

Chapter 3

FrameMaker Event Names

FrameMaker Events

IMPORTANT: Events are allowed only in Event Scripts.

The following table lists the special events names available under FrameMaker and ElmScript. It also describes when and under what circumstances they are run. Note also that one FrameMaker action might generate several events. This is especially true when documents are opened, closed or saved. If you wish to process an event, put the event with the corresponding name in your script. If you do not have an event with the corresponding name then your script won't be notified when the particular action occurs.

Some events occur in pairs. These are noted by the form `NotePreXXXX` and `NotePostXXXX`, where the `XXXX` represents the event. The `NotePreXXXX` event item is useful for when you might want to cancel the action via the `return` command. The `NotePostXXXX` event item provides the file name for file events.

FrameMaker Event parameters

All events are given the active document at the time of the event (`FrameDoc`). If the event is for a filter function, then the document object is the document into which the data is imported. If it is zero, then the filter must create a new document.

Some events are also passed a string value. Most of the time this value is the file name of the function in progress (`Filename`). This `Filename` is sometimes still an empty string when some events are run. Other times it is a string value containing some text, though the parameter is still called `Filename`. Other times there is no string value at all. It will be noted in the table below if the string value is used. Otherwise it is not available.

A third parameter (`IParm`) may also be passed to certain events. This is also noted in the table below. It is usually the object involved in the event.

Table 237: List of FrameMaker Events

Event Name	Event Description
<code>NoteBackToUser</code>	It is run after the FrameMaker product finishes processing the user action. This is a very <i>expensive</i> event for FrameMaker to process. By expensive that means that it takes up much resources, because this event is run everytime anything happens in the FrameMaker session. Use only when you absolutely need to.
<code>NoteBodyPageAdded</code>	It is run after the FrameMaker product adds the body page
<code>NoteBodyPageDeleted</code>	It is run after the FrameMaker product deletes the body page
<code>NoteClientCall</code>	When another API client calls your script.
<code>NoteCmsPostCheckoutItem</code>	FM 2015 or greater

Table 237: List of FrameMaker Events

Event Name	Event Description
NoteCmsPostOpenItem	FM 2015 or greater
NoteCmsPreCheckOutItem	FM 2015 or greater
NoteCmsPreOpenItem	FM 2015 or greater
NoteDirtyBook	It is run after a book is changed
NoteDirtyDoc	It is run after a document is changed
NoteDisplayClientTiDialog	It is run after the user clicked an inset
NoteDisplayClientXRefDialog	It is run after the user selects the Cross reference dialog. FM 9.0 or greater
NoteFilterFiletoFile	Filter a file on import or export via a file to file filter.
NoteFilterIn	It is run before checking the type of the file to be opened
NoteFilterOut	It is run before a document is saved
NoteGraphicPropChanged	FM 2015 or greater
NoteMathMLChangeNsPrefix	FM 12.0 or greater
NoteMathMLDbClicked	FM 12.0 or greater
NoteMathMLDeselected	FM 12.0 or greater
NoteMathMLGenerateImage	FM 12.0 or greater
NoteMathMLPopulateObj	FM 12.0 or greater
NoteMathMLSelected	FM 12.0 or greater
NotePostActiveDocChange	
NotePostAutoSaveDoc	It is run after saving a document. Filename is the name of the file.
NotePostBookComponentOpen	It is run after opening a file. Filename is the name of the file.
NotePostChangeElement	It is run after the FrameMaker product changes a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePostCreateMML	FM 12.0 or greater
NotePostCopyElement	It is run after the FrameMaker product copies a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePostDistill	It is run after distilling the PostScript
NotePostDragElement	It is run after the FrameMaker product drags a structural element. Filename is the element tag. IParm is the Element ID.
NotePostExport	It is run after exporting a text or graphic file. Filename is the name of the exported file.
NotePostFileType	It is run after checking the type of the file to be opened

Table 237: List of FrameMaker Events

Event Name	Event Description
NotePostFunction	It is run after the FrameMaker product executes command or adds text to the document. Filename is the text typed in by the user. IParm is the f-code for the function.
NotePostGenerate	It is run after generating a file. Filename is the name of the file.
NotePostGoToXRefSrc	It is run after the FrameMaker product goes to the cross-reference source. The IParm is the cross-reference object.
NotePostHypertext	It is run after the FrameMaker product executes the hypertext command
NotePostImport	It is run after importing a text or graphic file. Filename is the name of the imported file.
NotePostImportElemDefs	It is run after the FrameMaker product imports element definitions. Filename is the element tag. IParm is the f-code for the function.
NotePostInlineTypeIn	It is run after double-byte text entry
NotePostInsertElement	It is run after the FrameMaker product inserts a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePostMouseCommand	It is run after the FrameMaker product responds to the mouse click
NotePostOpenBook	It is run after opening the binary Book file. Filename is the name of the file.
NotePostOpenBookMIF	It is run after opening the Book Mif file. Filename is the name of the file.
NotePostOpenDoc	It is run after opening the binary document file. Filename is the name of the file.
NotePostOpenMIF	It is run after opening the Mif file. Filename is the name of the file.
NotePostOpenSGML	It is run after opening the sgml file. Filename is the name of the file.
NotePostOpenXML	It is run after opening the xml file. Filename is the name of the file.
NotePostPrint	It is run after the FrameMaker product prints a document or book. Filename is the name of the file.
NotePostQuitBook	It is run after closing a book. Filename is the name of the file.
NotePostQuitDoc	It is run after closing a document. Filename is the name of the file.
NotePostQuitSession	Immediately before exiting the session

Table 237: List of FrameMaker Events

Event Name	Event Description
NotePostRevertBook	It is run after reverting a book. Filename is the name of the file.
NotePostRevertDoc	It is run after reverting a document. Filename is the name of the file.
NotePostSaveAsPDFDialog	It is run after specifying Acrobat settings and generating PostScript.
NotePostSaveBook	It is run after saving a book. Filename is the name of the file.
NotePostSaveBookMIF	It is run after saving a Book MIF file. Filename is the name of the file.
NotePostSaveDoc	It is run after saving a binary document. Filename is the name of the file.
NotePostSaveMIF	It is run after saving a MIF file. Filename is the name of the file.
NotePostSaveSGML	It is run after saving a sgml file. Filename is the name of the file.
NotePostSaveXML	It is run after saving a xml file. FM 7.0 or greater Filename is the name of the file.
NotePostSetAttrValue	It is run after the FrameMaker product sets an attribute value. Filename is the attribute name. IParm is the Element ID.
NotePostUpdateXRefs	It is run after the FrameMaker product updates cross references.
NotePostWrapElement	It is run after the FrameMaker product inserts a structural element. Filename is the element tag. IParm is the Element ID.
NotePreAutoSaveDoc	It is run before saving a document. Filename is the name of the file.
NotePreBookComponentOpen	It is run before opening a file component. Filename is the name of the file.
NotePreChangeElement	It is run before the FrameMaker product changes a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePostCreateMML	FM 12.0 or greater
NotePreCopyElement	It is run before the FrameMaker product copies a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePreDistill	It is run before distilling the PostScript

Table 237: List of FrameMaker Events

Event Name	Event Description
NotePreDragElement	It is run before the FrameMaker product drags a structural element. Filename is the element tag. IParm is the Element ID.
NotePreExport	It is run before exporting a text or graphic. Filename is the name of the exported file.
NotePreFileType	It is run before checking the type of the file to be opened
NotePreFunction	It is run before the FrameMaker product executes command or adds text to a document. Filename is the text typed in by the user. IParm is the f-code for the function.
NotePreGenerate	It is run before generating a file
NotePreGoToXRefSrc	It is run before the FrameMaker product goes to the cross-reference source. The IParm is the cross-reference object.
NotePreHypertext	It is run before the FrameMaker product executes the hypertext command. Filename is the hypertext command.
NotePreImport	It is run before importing a text or graphic. Filename is the name of the imported file.
NotePreImportElemDefs	It is run before the FrameMaker product imports element definitions. Filename is the element tag. IParm is the f-code for the function.
NotePreInlineTypeIn	It is run before the double-byte text entry
NotePreInsertElement	It is run before the FrameMaker product inserts a structural element. Filename is the element tag. IParm is the f-code for the function.
NotePreMouseCommand	It is run before the FrameMaker product responds to the mouse click
NotePreOpenBook	It is run before opening a binary book file. Filename is the name of the file.
NotePreOpenBookMIF	It is run before opening a book mif file. Filename is the name of the file.
NotePreOpenDoc	It is run before opening a binary document. Filename is the name of the file.
NotePreOpenMIF	It is run before opening a mif document file. Filename is the name of the file.
NotePreOpenSGML	It is run before opening a sgml document file. Filename is the name of the file.
NotePreOpenXML	It is run before opening a xml document file. FM 7.0 or greater Filename is the name of the file.

Table 237: List of FrameMaker Events

Event Name	Event Description
NotePrePrint	It is run after the user clicks OK in the Print dialog box, but before the FrameMaker product prints the document or book. Filename is the name of the file.
NotePreQuitBook	It is run before closing a book file. Filename is the name of the file.
NotePreQuitDoc	It is run before closing a document. Filename is the name of the file.
NotePreQuitSession	It is run before the OK to Exit dialog box appears
NotePreRevertBook	It is run before reverting a book. Filename is the name of the file.
NotePreRevertDoc	It is run before reverting a document. Filename is the name of the file.
NotePreSaveAsPDFDialog	It is run before specifying Acrobat settings and generating PostScript
NotePreSaveBook	It is run before saving a book. Filename is the name of the file.
NotePreSaveBookMIF	It is run before saving a Book MIF file. Filename is the name of the file.
NotePreSaveDoc	It is run before saving a document. Filename is the name of the file.
NotePreSaveMIF	It is run before saving the document as a MIF file. Filename is the name of the file.
NotePreSaveSGML	It is run before saving the document as a sgml file. Filename is the name of the file.
NotePreSaveXML	It is run before saving the document as a xml file. FM 7.0 or greater Filename is the name of the file.
NotePreSetAttrValue	It is run before the FrameMaker product sets an attribute value. Filename is the attribute name. IParm is the Element ID.
NotePreUpdateXRefs	It is run after the FrameMaker product updates cross references.
NotePreWrapElement	It is run before the FrameMaker product wraps a structural element. Filename is the element tag. IParm is the Element ID.
NoteUpdateAllClientTi	It is run when the client needs to update insets that belong to it. IParm is a flag indicator with one of the following values: UpdateAllAutomaticClientTi means update only those with TiAutomaticUpdate of True. UpdateAllManualClientTi means update only those with a TiAutomaticUpdate of False. UpdateAllClientTi means update all of the regardless of the TiAutomatic setting.
NoteUpdateClientTi	It is run when the client needs to update a specified inset. The IParm is the Text inset object.

Hypertext Event

Hypertext events are processed with a special event name called `Message`. If you wish to process this event, put in an event with the name `Message` in your script. If you do not have an event with the name `Message` then your script won't be notified when a Hypertext marker is pressed.

FrameMaker Message parameters

The `Message` hypertext event is passed three parameters. The first is the document object where the hypertext marker resides (`FrameDoc`). The next parameter is the `FrameMaker` object of the marker object that was pressed (`FrameObject`). The last parameter is a string variable containing the message in the hypertext command (`Message`).

Example:

```
. . .
Event Message
  MsgBox 'The user pressed the marker-'+FrameObject;
  MsgBox 'In document - '+FrameDoc;
  MsgBox 'With the message - '+ Message;
EndEvent
. . .
```

ElmScript Events

`ElmScript` has four events that it will run (if defined in the script) during event scripts. These are `Initialize`, `Terminate`, `RunCommandExec` and `AfterFrameInit`. The `Initialize` event is run when an event script is first installed. This is used primarily to initialize variables and to create menus and menu commands for the event script. Of course it may be used for whatever you wish. The `Terminate` event is run when an event script is uninstalled. This is a convenient place to remove menus, menu items and delete any `EsObjects`. The `RunCommandExec` event is run, if the user tries to 'Run' an event script instead of installing it. The usual purpose of this event is to inform to user that the script should have been installed.

The `AfterFrameInit` event is special event that is called at one specific time in a `FrameMaker` session. It is called after `FrameMaker` has initialized all other API clients (including `ElmScript`), after the `ElmScript` initial script has run, and when `FrameMaker` has finished initializing itself. Therefore, it will be run only for event scripts that were installed via the initial script facility. This event can be used for batch processing. It can be used to communicate with other clients/scripts that were not initialized earlier. It can also be used to handle changes (such as menu rearrangement) that cannot be done in the `Initialize` event, because `FrameMaker` is not yet completely initialized.

IMPORTANT: `FrameMaker` will only run the `AfterFrameInit` event if the `ElmScript` client has the `TakeControl` option in the client registration in the `FrameMaker` initialization file (`maker.ini`). The entry should look something like the following: `fsl=TakeControl, ElmScript, c:\Program Files\ElmSoft\ ...` The install program should do this for you.

Example:

```
Event Initialize
  New Menu . . . .
  Set initVarValue = 100;
  . . .
EndEvent

Event Terminate
  Remove MenuObject ....
EndEvent

Event AfterFrameInit

EndEvent

Event RunCommandExec
  MsgBox 'This script cannot be run. It must be installed';
EndEvent
```

ElmScript Menu Events

The last type of event is the ElmScript menu events. These are events that the scriptwriter defines for processing menu items created within the script.

Example:

```
. . .
New Command Label('My command') Addto('FileMenu') EventProc(MyEvent);
. . .
Event MyEvent
  MsgBox 'The user press my menu command';
EndEvent
. . .
```


Chapter 4

F - Codes

FrameMaker Codes

Fcodes are predefined codes that specify user actions such as menu item clicks, keyboard commands, cursor movements and text entry. You may use the **Execute Fc** ElmScript command to simulate one or more of these actions

The following table describes the F-codes available for the **Execute Fc** command. The documentation available for these f-codes is minimal. The description column will contain the available information. Sometimes it will include the keyboard shortcut, sometimes a complete description and sometimes the name of the f-code itself is all that's available.

When keyboard shortcut equivalents are supplied, the ! symbol is for the Escape key. This is platform independent. The ^ symbol is the control key and s means shift.

Example:

To move the cursor to the beginning of the current line and then highlight the next sentence:

```
Execute Fc CsrBol HighSent;
```

Table 238: List of F-codes

F-code Identifiers	Description of F-code
CmdOpenDitamapComponent	FM10 or greater
CharApplyCatToSel	
CharApplyTag	Set char tag from menu
CharDesignCat	
CharDesignKitApply	
CharDesignKitNewFormat	
CharDesignKitUpdateAll	
CharDesignKitUpdateOptions	
CondTextKitApply	
CsrBeforeElement	!sB - Before beginning of element
CsrBoe	!sS - Beginning of element
CsrBol	^a - Beginning of line
CsrBop	~[- Previous paragraph start
CsrBos	~a - Previous sentence start
CsrBot	~} - End of flow

Table 238: List of F-codes

F-code Identifiers	Description of F-code
CsrBotr	~Z,+[End] - Bottom of column
CsrBow	~b - Previous word start
CsrDown	^n, [Down],
CsrEoe	!sE - End of element
CsrEol	^e - End of line
CsrEop	~] - Next paragraph end
CsrEos	~e - Next sentence end
CsrEow	~f - Next word end
CsrFirstCol	!bf - First text column on current page
CsrHome	~z,+[Home], - Top of column
CsrIntoChild	!sN - Into child
CsrLeft	^b, [Left]
CsrNextBop	!bp - Next paragraph start
CsrNextBos	!bs - Next sentence start
CsrNextBow	!bw - Next word start
CsrNextCol	!bn - Next text column on current page
CsrNextElement	!sD - Next element/snippet start or end
CsrPrevElement	sU - Previous element/snippet start or end
CsrRight	^f, [Right]
CsrTop	~{ - Beginning of flow
CsrUp	^p, [Up]
DelBol	^u - Delete backward to start of line
DelBOS	!ka - Delete backward to previous sentence end
DelBow	~[Delete], ~[BackSpace], ~^h - Delete
DelCharBwd	[Delete], [BackSpace], ^h - Delete
DelCharFwd	^d - Delete forward one character
DelEol	^k - Delete forward to end of line
DelEos	~k - Delete forward to next sentence end
DelEow	~d - Delete forward to next word start
DelNextSS	!ks - Delete forward to next sentence start
DelSel	see DelCharBWD, Clear shortcut
DelWordEnd	!kf - Delete forward to next word end
DelWordStart	!kb - Delete backward to previous word start
ElemBam	!eer - Repeat last insert/wrap/change

Table 238: List of F-codes

F-code Identifiers	Description of F-code
ElemChangeCatAtSel	change from element catalog.
ElemInsCatAtSel	Insert from element catalog.
ElemWrapCatAtSel	Wrap from element catalog.
ElemInsertCustom1	Insert first custom element.
ElemInsertCustom2	Insert second custom element.
ElemInsertCustom3	Insert third custom element.
ElemInsertCustom4	Insert fourth custom element.
ElemInsertCustom5	Insert fifth custom element.
ElemInsertCustom6	Insert sixth custom element.
ElemInsertCustom7	Insert seventh custom element.
ElemInsertCustom8	Insert eighth custom element.
ElemInsertCustom9	Insert ninth custom element.
ElemInsertCustom10	Insert tenth custom element.
ExtendSel	Text Selection
FMRptCmdsByShortcut	
FMRptCmdsByTag	
FMSeparator	Menu item Separator
Fmterminate	Quit FrameMaker
FocusInputAttrEditor	!Fia - Attribute editor kit
FocusInputCellFmt	Obsolete?
FocusInputCond	!Fio - Conditional text
FocusInputCustRS	!Fir - Custom ruling and shading
FocusInputDoc	!Fid - Current document
FocusInputElemCtx	!Fie - Show Element Context kit
FocusInputElemCtxAv	FM11 or greater
FocusInputFontFmt	!Fic - Character format
FocusInputMarker	!Fim - Marker
FocusInputPgffmt	!Fip - Paragraph format
FocusInputSearch	!Fif - Find
FocusInputSpell	!Fis - Spelling
FocusInputStrWin	!Fiv - Structure window
FocusInputTblfmt	!Fit - Table format
FocusInputValidation	!Fiw - Validation Kit
HighChar	!hc - Highlight next character

Table 238: List of F-codes

F-code Identifiers	Description of F-code
HighCharNext	!Hc - Move active end of selection fwd 1 char
HighCharPrev	!HC - Move active end of selection back 1 char
HighClear	!h0 - Clear Selection
HighColBot	!hm - Extend to bottom of column
HighColTop	!ht - Extend to top of column
HighElement	!hE - Highlight next element
HighElementNext	!hN - Move active forward back 1 element
HighElementPrev	!hP - Move active end back 1 element
HighFlow	!ea - Select all in flow/page/table/frame
HighFlowBeg	!hg - Extend to beginning of flow
HighFlowEnd	!hn - Extend to end of flow
HighLine	!hl - Highlight next line
HighLineBeg	Windows specific - Select line to the beginning
HighLineDown	!hd - Extend one line down
HighLineEnd	Windows specific - Select line to the end
HighLineNext	!HL - Move active end forward 1 line (select a whole line)
HighLinePrev	!HL - Move active end back 1 line (select a whole line)
HighLineUp	!hu - Extend one line up
HighParent	!heP - Extend to all of parent
HighPgf	!hp - Highlight next paragraph
HighPgfNext	!Hp - Move active end forward 1 paragraph
HighPgfPrev	!HP - Move active end back 1 paragraph
HighSameCB	!hF - Select text with the same char fmt
HighSameCond	!hC - Select text with the same conditions
HighSent	!hs - Highlight next sentence
HighSentNext	!Hs - Move active end forward 1 sentence
HighSentPrev	!HS - Move active end back 1 sentence
HighSHL	!hb - Shift highlighting left 1 char
HighSHR	!hf - Shift highlighting right 1 char
HighSiblings	!hS - Extend to all of parent's contents
HighWord	!hw - Highlight next word
HighWordNext	!Hw - Move active end forward 1 word
HighWordPrev	!HW - Move active end back 1 word
InLineTypeIn	Inline input for Asian text

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdAbort	^c - Abort a long process like import
KbdAbortDlgS	Special for X and FrameServer
KbdAcceptAllChange	
KbdAcceptChange	
KbdAcceptChangeAndNext	FM11 or greater
KbdAcrobatSetup	!oda - Acrobat Setup - Format Menu - Document
KbdActivateInPdfByDefault	FM11 or greater
KbdAddAutoCorr	!lac - add to auto corrections - Spell Checker
KbdAddDocDict	!lad - add to document dict - Spell Checker
KbdAddMarkerType	?? - Add Marker Type by name FM 6.0 Only
KbdAddPage	!spa - Other Page Commands
KbdAddUsrDict	!lap - add to personal dict - Spell Checker
KbdAlign	!ga - Graphic Menu Command
KbdAlignBottom	!jb - Bottom align
KbdAlignMiddle	!jm - Top/bottom (middle) align
KbdAlignTop	!jt - Top align
KbdAllCap	~u - convert selected text to cap - Edit Menu
KbdAllCapH	like ALLCap, but keep text selected - Edit Menu
KbdAllLower	~l - convert selected text to lower case - Edit Menu
KbdAllLowerH	like ALLLower, but keep text selected - Edit Menu
KbdAllSelect	!ls - smart selection
KbdAnchor	!sa - Special Menu Command
KbdAntiPutInLine	!me - Expand (unwrap)
KbdArcTool	!la - Tools Window Command
KbdAscAddDocDict	FM10 or greater
KbdAscAddUsrDict	FM10 or greater
KbdAscCorrect	FM10 or greater
KbdAscEnableAutoSpellCheck	FM12 or greater
KBDAssignElementId	FM10 or greater
KbdAtomizeInset	!gU - Ungroup FmVect import.
KbdAttrCond	
KbdAttrConfigFileMaker	FM10 or greater
KbdAttrDispOpts	!vA -Attribute Display
KbdAttrEditQuick	^7 - attribute edit quick key

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdBack	!gb - Graphic Menu Command
KbdBackStack	!vP - Backward Link
KbdBackTab	+ [Tab] - Tab backwards in a dialog
KbdBannerText	FM11 or greater
KbdBodyPage	!vB - Other Page Commands
KbdBookAddFile	!ff - Add file to book
KbdBookAddLinkedFolder	FM 2017 or greater
KbdBookCloseSelectedFiles	FM 6.0 Only
KbdBookDisplayFilename	FM 6.0 Only
KbdBookDisplayText	FM 6.0 Only
KbdBookDeleteFile	FM 6.0 Only
KbdBookEditDefine	!fd - Set up file in book
KbdBookEditFilelist	!fe - Rearrange files in book Obsolete for FM 6.0
KbdBookMoveFileUp	FM 6.0 Only
KbdBookMoveFileDown	FM 6.0 Only
KbdBookOpenSelectedFiles	FM 6.0 Only
KbdBookPrintSelectedFiles	FM 6.0 Only
KbdBookRenameFile	!fe - Rearrange files in book. FM 6.0 Only
KbdBookSaveSelectedFiles	FM 6.0 Only
KbdBorders	!vb - View Menu Command
KbdBrowseURL	FM10 or greater
KbdCacheStats	print cache statistics
KbdCapital	!eC - Edit Menu
KbdCapture	!ftP - Capture portion of screen, not used
KbdCaptureLiveDialog	!dc - capture (print) live dialog
KbdCBarPro	!ob - Change Bars - Format Menu - Document
KbdChangeDict	!lcd - change dictionaries - Spell Checker
KbdChangeQuick	!Ec, ^3 - change element quick key
KbdCharfmtDelete	
KbdCharfmtDeleteUnused	FM10 or greater
KbdCheckBatch	!lb - batch spell check - Spell Checker
KbdCheckDoc	!le - check entire doc - Spell Checker
KbdCheckPage	!lp - check page - Spell Checker
KbdCheckSel	!ls - check selection - Spell Checker

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdClear	!eb - Edit Menu
KbdClearAuto	!lca - clear auto corrections - Spell Checker
KbdCLOPWin	Close/Open Window - Window Menu Command
KbdCloseAttrEditor	!Ca - Attribute editor kit
KbdCloseCharacterCatalog	
KbdCloseCharacterDesigner	
KbdCloseCommandPalette	
KbdCloseCond	!Co - Conditional text
KbdCloseCustRS	!Cr - Custom ruling and shading
KbdCloseElemCtx	!Cx - Close Element Context kit
KbdCloseElemCtxAv	FM11 or greater
KbdCloseElementCatalog	
KbdCloseEquationsPalette	
KbdCloseFontFmt	!Cc - Character format
KbdCloseHypertext	!Ch - Close Hypertext kit
KbdCloseMarker	!Cm - Marker
KbdCloseParagraphCatalog	
KbdCloseParagraphDesigner	
KbdClosePgffmt	!Cp - Paragraph format
KbdCloseSearch	!Cf - Search
KbdCloseSpell	!Cs - Spelling
KbdCloseStrWin	!Cv - Structure window
KbdCloseTableCatalog	FM10 or greater
KbdCloseTableDesigner	
KbdCloseTblfmt	!Ct - Table format
KbdCloseToolsPalette	
KbdCloseTopPalette	
KbdCloseValidation	!Cw - Validation Kit
KbdCloseWin	!wc - Window Menu Command
KbdCmsConnectionMgr	FM10 or greater
KbdCmsPreference	FM10 or greater
KbdCmsUploadDocOrBook	FM10 or greater
KbdCollLayout	!ocl - Format Menu - Page Layout
KbdColor	!vcd - View Menu Command

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdCombinedFonts	!ocf - Combined Fonts - Format Menu - Document
KbdCommandPalette	!vq - Quick Access Bar
KbdCompare	!ftc - Compare docs or books - File Menu (Shift)
KbdCondEdit	
KbdCondIndicator	
KbdCondInQuick	!qC - Conditional text “In” quick key
KbdCondNotInQuick	!qD - Conditional text “NotIn” quick key
KbdCondText	!sC - Special Menu Command
KbdCondToggleOverR	!vO - View Menu Command
KbdCondVisibility	!vC - View Menu Command
KbdCondVisOnlyQuick	!qS Show One Conditional Text Tag
KbdConfigureTcColor	FM10 or greater
KbdConnectTextFrame	!CC -Connect text frames - Format Menu - Cust. Page Layout
KbdConstrain	- Graphic Menu Command
KbdContextMenu	MS Windows Only
KbdCopy	~w, !ec - Edit Menu
KbdCopyAttrs	!eA - Copy Element Attributes
KbdCopyColW	!eyw - Copy Column Width
KbdCopyCond	!eyd - Copy Condition Setting
KbdCopyFont	!eyc - Copy font properties - Edit Menu
KbdCopyPgf	!eyp - Copy pgf properties - Edit Menu
KbdCorrect	!lcw - correct word - Spell Checker
KbdCreateMultiMediaLink	FM11 or greater
KbdCreateMultiMediaLinkTbl	FM11 or greater
KbdCreatePublisher	Mac only - Edit Menu
KbdCSHelpMode	~?, [F1]
KbdCurTail	!CN - Cut Next - Format Menu - Cust. Page Layout
KbdCustomNew	- Custom Blank Paper dialog
KbdCustTextFrame	!ocf - Customize text frame - Format Menu - Cust. Page Layout
KbdCut	^w, !ex - Edit Menu
KbdCutBoth	!CB - Cut Both - Format Menu - Cust. Page Layout
KbdCutHead	!CP - Cut Previous - Format Menu - Cust. Page Layout
KbdDash0	!0d- select first dashed pattern
KbdDash1	!9d- select last dashed pattern

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdDashOption	!Idi - bring up dashed options dialog
KbdDecDash	!+d- select previous dashed pattern
KbdDecFill	!-f - Decrement fill pattern - Fill Pattern Command
KbdDecPen	!-p - Decrement pen pattern - Pen Pattern Command
KbdDecWidth	!-w - Decrement line width - Line Width Command
KbdDelDocDict	!lxd - del from document dict - Spell Checker
KbdDeleteInlineToc	FM 2015 or greater
KbdDeletePage	!spd - Other Page Commands
KbdDelinkHotSpot	FM11 or greater
KbdDelMarkerType	?? - Delete Marker Type by name - Special Menu Command
KbdDelUsrDict	!lxp - del from personal dict - Spell Checker
KbdDeRefRef	dereference ref @ ip
KbdDirectionLtr	FM 2015 or greater
KbdDirectionRtr	FM 2015 or greater
KbdDistribute	!gd - Graphic Menu Command
KbdDocInfo	PDF Doc info. FM 6.0 Only
KbdDocReport	!ftr - Document reports
KbdDocumentNum	!odn - Document Numbering - Format Menu - Document
KbdDontHyphen	!ns, ~ - suppress hyphenation
KbdDoubleQuote	“ Smart Quotes
KbdDreModeM	!dm - make Mac the current view
KbdDreModeW	!dw - make Windows the current view
KbdDreModeX	!dx - make X the current view
KbdDSExit	exercise dsexit
KbdDumpHypertext	!dh - Dump h-text selection to file
KbdECapture	!ftp - Capture portion of screen, compressed
KbdAttributeEdit	!EA - Edit Attributes
KbdEditHotSpot	FM11 or greater
KbdEditLinks	MS Windows Only
KbdEditMarkerType	!emt - Edit Marker Types. FM 6.0 Only
KbdEditMathMLEquation	FM12 only-Removed in FM 2015
KbdElemBorder	!vE - Element borders.
KbdElemCatOpts	!EOC - Set available elements.
KbdElemDemote	!ED - Demote element.

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdElementWin	!EC - element catalog window
KbdElemMerge1st	!Em - Merge.
KbdElemMergeLast	!EM - Merge into last.
KbdElemPromote	!EP - Promote element.
KbdElemSplit	!Es - Split.
KbdElemTags	!vT - Element boundaries as Tags.
KbdElemTransposeNext	!ET - Transpose with next.
KbdElemTransposePrev	!Et - Transpose with Previous.
KbdElemUnwrap	!Eu - Unwrap.
KbdEMSpace	![Space]m - em space = 1 em
KbdENSpace	![Space]n - en space = 1/2 em
KbdEquation	!pe - Equation Sizes- Special Menu Command
KbdEvacuate	(unbound) Force assertion botch
KbdExportDocument	Export document from menu
KbdExportGraphic	Export Graphic from menu
KbdExposeWin	!we - Window Menu Command
KbdFill0	!0f - Set to “first” fill pat - Fill Pattern Command
KbdFill1	!9f - Set to last fill pattern - Fill Pattern Command
KbdFindNext	
KbdFindPrev	
KbdFirstPage	!pf, ~<, +[F6] - Other Page Commands
KbdFirstTab	^Tab - First tab (as in dialogs)
KbdFlipLR	!gh - Graphic Menu Command
KbdFlipUD	!gv - Graphic Menu Command
KbdFmPip	FM10 or greater
KbdFNext	~s, !fin - Search forward
KbdFontDesign	!ocd - Character Format Designer - Format Menu
KbdFontQuick	!qc, F8, ^8 - Char fmt quick key
KbdFontWin	!occ - font catalog window
KbdFootnote	!sf - Special Menu Command
KbdFootnotePro	!of - Footnote Properties - Format Menu - Document
KbdFPrev	~r, !fip - Search backward
KbdFrameTool	!Im - Tools Window Command
KbdFreeTool	!If - Tools Window Command

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdFreeze	!pz - Other Page Commands
KbdFront	!gf - Graphic Menu Command
KbdFullRulers	!oa - toggle full/abbreviated rulers
KbdGenerate	!fg - File Menu
KbdGetTrigger	^] - Get trigger for recorded keystrokes
KbdGoTopage	!vp, ^g - Other Page Commands
KbdGraphicSetName	FM11 or greater
KbdGravity	!gy - Graphic Menu Command
KbdGravity0	Graphics Palette
KbdGravity1	Graphics Palette
KbdGrid	!vg - View Menu Command
KbdGroup	!gg - Graphic Menu Command
KbdHardHyphen	!-h, ~- - nonbreaking hyphen
KbdHardReturn	~[Return] - Hard return
KbdHardSpace	![Space]h, ^[Space] - hard space (not word delimiter)
KbdHeatRef	heat reference @ ip
KbdHelp	!fh
KbdHelpIndex	- Help Index
KbdHelpKeys	!?k - Keyboard Shortcut Help
KbdHelpKyle	- Help about Kyle FM 6.0 Only
KbdHelpOnlineManuals	- FrameMaker Online Manuals
KbdHelpOverview	- FrameMaker Overview
KbdHelpSamples	- Samples and Clip Art Help
KbdHeroicOpen	!oH
KbdHideBorders	FM 6.0 Only
KbdHideCondInd	FM 6.0 Only
KbdHideElemBorder	FM 6.0 Only
KbdHideGraphics	FM 6.0 Only
KbdHideGrid	FM 6.0 Only
KbdHideHotSpotIndicators	FM11 or greater
KbdHideLinkBoundaries	FM 6.0 Only
KbdHideRulers	FM 6.0 Only
KbdHideSymbols	FM 6.0 Only
KbdHideWin	!wh - Window Menu Command

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdHISHWin	Hide/show toggle - Window Menu Command
KbdHotSpot	FM11 or greater
KbdHotSpotIndicators	FM11 or greater
KbdHotSpotSPod	FM11 or greater
KbdHist	Shows the undo/redo history palette. (!eh, [Hist]) <i>Frame 7.2 or greater</i>
KbdHScroll	
KbdHypertext	!sh - Special Menu Command
KbdHyphenate	!l- - hyphenate word - Spell Checker
KbdHypertextShtcut	
KbdImageImport	FM 2017 or greater
KbdImport	!fif - Import File - File Menu
KbdIncDash	!-d- select next dashed pattern
KbdIncFill	!+f - Increment fill pattern - Fill Pattern Command
KbdIncPen	!+p - Increment pen pattern - Pen Pattern Command
KbdINCWidth	!+w - Increment line width - Line Width Command
KbdIndentList	Tab for indent and outdent list FM 2017 or greater
KbdIndexAuthor	FM 6.0 Only
KbdIndexMarker	FM 6.0 Only
KbdIndexReferences	FM 6.0 Only
KbdIndexStandard	FM 6.0 Only
KbdIndexSubject	FM 6.0 Only
KbdInfo	^[F1]
KbdInitCap	~c - convert selected text to initial caps - Edit Menu
KbdInitCapH	like InitCap, but keep text selected - Edit Menu
KbdInlineAttrEdtr	FM11 or greater
KbdInlineToc	FM 2015 or greater
KbdInsertObject	MS Windows Only
KbdInsertQuick	!Ei, ^1 - insert element quick key
KbdInset	!si - Special Menu Command
KbdItem1stLogical	
KbdItemDown	
KbdItemFirstFocus	
KbdItemLeft	
KbdItemNextLogical	

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdItemNextPhysical	
KbdItemPrevLogical	
KbdItemPrevPhysical	
KbdItemRight	
KbdItemSelect	
KbdItemUp	
KbdJoinCurves	!gj - Graphic Menu Command
KbdKbMacro	!ftk - File Menu
KbdKeepTool	!gk - Graphic Menu Command
KbdKernDown	^[Down] - Move 1 point down
KbdKernDown6	+^[Down] - Move 6 point down
KbdKernHome	^[Home] - Move back to baseline
KbdKernLeft	^[Left] - Move 1 point left
KbdKernLeft6	+^[Left] - Move 6 point left
KbdKernRight	^[Right] - Move 1 point right
KbdKernRight6	+^[Right] - Move 6 point right
KbdKernUp	^[Up] - Move 1 point up
KbdKernUp6	+^[Up] - Move 6 point up
KbdLastPage	!pl, ~>, +[F7] - Other Page Commands
KbdLastTool	!l1 select last-used tool - Tools Window Command
KbdLgEqn	!ml - Large equation
KbdLineLayout	!oll - Format Menu - Page Layout
KbdLineNumPro	FM11 or greater
KbdLineNumToggle	FM11 or greater
KbdLineTool	!l1 - Tools Window Command
KbdLinkBoundaries	Turn link borders on and off. - Edit Menu
KbdListFigure	FM 6.0 Only
KbdListMarker	FM 6.0 Only
KbdListMarkerAlpha	FM 6.0 Only
KbdListPgf	FM 6.0 Only
KbdListPgfAlpha	FM 6.0 Only
KbdListReferences	FM 6.0 Only
KbdListTable	FM 6.0 Only
KbdManCond	

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdMarkers	!sm - Special Menu Command
KbdMasterPage	!vM - Other Page Commands
KbdMathMLEdit	FM 2015 or greater
KbdMathWin	!se, !mw - equations window
KbdMaximize	MS Windows Only
KbdMedEqn	!mm - Medium equation
KbdMemFail	exercise memfail
KbdMemStats	print busy/free memory totals
KbdMenuBarFocus	simulates Menubar Focus for Bookkit
KbdMenuComplete	!vmc - View Menu Command
KbdMenuCustom	!vmu - View Menu Command
KbdMenuModify	!vmm - View Menu Command
KbdMenuQuick	!vmq - View Menu Command
KbdMenuReset	!vmr - View Menu Command
KbdMinimize	MS Windows Only
KbdModeRotateTool	Rotate selected object. FM 6.0 Only
KbdMoveWin	!wm - Window Menu Command
KbdNavigateDown	Navigate down in simplified xml. FM 2017 or greater
KbdNew	!fn- File Menu
KbdNewBook	!fN FM 6.0 Only
KbdNewHypertext	!mh - Insert new hypertext - Special Menu Command
KbdNewMarker	!mk - Insert new marker - Special Menu Command
KbdNewMaster	!omp - Format Menu - Page Layout
KbdNewProject	FM 2017 or greater
KbdNewVar	new variable @ ip
KbdNewXml	FM10 or greater
KbdNextPage	!pn, ^v, F7 - Other Page Commands
KbdNoChangeDB	+ [F8] - Set all items to As Is in dialog
KbdNormalizeTags	!ftf - Catalogize pgf and char formats
KbdNumberUtilityFtoN	FM 2017 or greater
KbdNumberUtilityItoN	FM 2015 or greater
KbdNumberUtilityNtoF	FM 2017 or greater
KbdNumberUtilityNtoI	FM 2015 or greater
KbdNumbering	!odn - Doc Numbering, !en Book. FM 6.0 Only

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdNumLock	NumLock key - change keypad between function pad and keypad.
KbdNumSpace	![Space]1 - number space
KbdOBAlignBottom	- Bottom align
KbdOBAlignCenter	- Center paragraph
KbdOBAlignLeft	- Left justify paragraph
KbdOBAlignMiddle	- Top/bottom (middle) align
KbdOBAlignRight	- Right justify paragraph
KbdOBAlignTop	- Top align
KbdObjDown	- Move 1 point down
KbdObjLeft	- Move 1 point left
KbdObjProps	!go - Graphic Menu Command
KbdObjRight	- Move 1 point right
KbdObjSelect	!lo - object selection
KbdObjSelectNoPref	shifted KbdObjSelect
KbdObjUp	- Move 1 point up
KbdOpen	^x^f, !fo - File Menu
KbdOpenAll	!fO - Open all - File Menu (Shift)
KbdOpenLine	^o - Open line
KbdOpenRepository	FM10 or greater
KbdOpenWin	!wo - Window Menu Command
KbdOptions	!vo - View Menu Command
KbdOvalTool	!le (“ellipse”) - Tools Window Command
KbdOutdentList	Tab for outdent list FM 2017 or greater
KbdPageBack	!omu - Format Menu - Page Layout
KbdPageBreak	!spb - Special Menu Command
KbdPageSetup	page setup for Mac - File Menu
KbdPageSize	!ops - Format Menu - Page Layout
KbdPageStatus	
KbdPageUpdate	!oup - Format Menu - Page Layout
KbdPagingation	FM 6.0 Only
KbdPaste	!ep - Edit Menu
KbdPasteSpecial	
KbdPopupMenu	
KbdPause0x01	Causes FrameMaker to sleep for 0.01 second

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdPause0x10	Causes FrameMaker to sleep for 0.1 seconds
KbdPause1x00	Causes FrameMaker to sleep for 1 second.
KbdPen0	!0p - Set to “first” pen pat - Pen Pattern Command
KbdPen1	!9p - Set to last pen pattern - Pen Pattern Command
KbdPgfDesign	!opd - Paragraph Format Designer - Format Menu
KbdPgffmtDelete	
KbdPgffmtDeleteUnused	FM10 or greater
KbdPgfQuick	!qp, F9, ^9 - Pgf fnt quick key
KbdPgfWin	!opc - paragraph catalog window
KbdPickObjProps	!gO - Graphic Menu Command
KbdPolyGTool	!1pg - Tools Window Command
KbdPolyLTool	!1pl - Tools Window Command
KbdPreviewAcceptAll	
KbdPreviewFBA	FM10 or greater
KbdPreviewOff	
KbdPreviewRejectAll	
KbdPrevPage	!pp, ~v, F6 - Other Page Commands
KbdPrint	!fp - File Menu
KbdPrintSetup	MS Windows Only
KbdProjectAddLocation	FM 2017 or greater
KbdProjectCreateFolder	FM 2017 or greater
KbdProjectDeleteFile	FM 2017 or greater
KbdProjectEditFile	FM 2017 or greater
KbdProjectExploreFile	FM 2017 or greater
KbdProjectManagerView	FM 2017 or greater
KbdProjectRenameFile	FM 2017 or greater
KbdProjectShowFilename	FM 2017 or greater
KbdProjectShowFilepath	FM 2017 or greater
KbdPutInLine	!mp - Shrinkwrap
KbdQABHelp	MS Windows Only
KbdQAccessToggle	MS Windows Only
KbdQuitAll	!fQ - Quit all - File Menu (Shift)
KbdQuitWin	!fq, !fc, ^x^c - Window Menu Command
KbdRAndF	!ra - Replace and find again

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdRecord	^] - Record keystrokes
KbdRedo	Performs a Edit->Redo !er, [Redo]. <i>Frame 7.2 or greater</i>
KbdRectTool	!lr - Tools Window Command
KbdReference	!sc - Special Menu Command
KbdReformatDoc	!RR - reformat entire document - Spell Checker
KbdRefPage	!vR - Other Page Commands
KbdRefreshDitamapRmView	Ditamap Toolbar command added. (FM10 or greater)
KbdRefreshWin	!wr, ^l - Window Menu Command
KbdRejectAllChange	
KbdRejectChange	
KbdRejectChangeAndNext	FM11 or greater
KbdRemovePoster	!frp Remove the poster set on the inset (FM10 or greater)
KbdRemoveStruct	
KbdSymFont	temporarily set IP font to Symbol
KbdRenameFrame	not bound currently
KbdRenameMarkerType	?? - Rename Marker Type - Special Menu Command
KbdRenameOrPlain	FM 6.0 Only
KbdRenamePage	!pN - rename master/reference page
KbdRepeat	!eh, [Repeat] (FM10 or greater)
KbdRepeatNew	- Repeat last new command - File Menu (Shift)
KbdReRotate	!gl - Graphic Menu Command
KbdResetDB	+ [F9] - Reset dialog
KbdReshape	!gr - Graphic Menu Command
KbdResizeBox	window resize box
KbdResizeBoxM	window resize box using ctrl-middle mouse
KbdRestore	MS Windows Only
KbdRestoreFont	restore IP Font
KbdRestoreSession	FM12 or greater
KbdReturn	[Return] - normal return
KbdRevert	!fr - File Menu
KbdReWrapInlineMath	!rwm - Shrink-wrap all inline math.
KbdRGlobal	!rg - Change all
KbdRibbonBar	MS Windows Only
KbdRMode	!dr - Remove platform dialog

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdROnce	^%, !ro - Change
KbdRotate	!gt - Graphic Menu Command
KbdRotateAgain	!gx - Graphic Menu Command
KbdRotateCCW	!gt - rotate 90 counterclockwise - Graphic Menu Command
KbdRotateCCWsmall	- rotate 15 or 90 counterclockwise - Graphic Menu Command
KbdRotateCW	!g+ - rotate 90 clockwise - Graphic Menu Command
KbdRotateCWsmall	- rotate 15 or 90 clockwise - Graphic Menu Command
KbdRotateNatural	!g0 - Graphic Menu Command
KbdRotateReset0	!g9 - Graphic Menu Command
KbdRotMinus	!g+ - rotate 90 clockwise - Graphic Menu Command
KbdRotPageMinus	!po - Rotate page CounterC - Format Menu - Cust. Page Layout
KbdRotPageNorm	!pU - Un-rotate Page - Format Menu - Cust. Page Layout
KbdRotPagePlus	!pO - Rotate page clockwise - Format Menu - Cust. Page Layout
KbdRotPlus	!gt - rotate 90 counterclockwise - Graphic Menu Command
KbdRoundRect	!IR - Tools Window Command
KbdRubiProps	FM 6.0 Only
KbdRulers	!vr - View Menu Command
KbdRunBBox	!gW - Bounding box runaround - Graphic Menu Command
KbdRunContour	!gw - Contour runaround - Graphic Menu Command
KbdRunGap	- Graphic Menu Command
KbdRunoff	!gq - Turn off runarounds - Graphic Menu Command
KbdRunProps	FM 6.0 Only
KbdSave	^x^s, !fs - File Menu
KbdSaveAll	!fS - Save all - File Menu (Shift)
KbdSaveAS	^x^w, !fa - File Menu
KbdSaveMeta	!Ftc Toggle mode so Save Text saves meta Chars
KbdSavesAs	so says cmdinit
KbdSaveAsPdfShare	FM10 or greater
KbdSaveAsPdfUbiq	FM10 or greater
KbdSaveAsXml	FM8 or greater
KbdScale	!gz - Graphic Menu Command
KbdSearch	^s, !ef - Edit Menu
KbdSearchCommand	FM 2017 or greater
KbdSearchRefs	FM10 or greater

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdSelectAll	!ea - Edit Menu
KbdSelectFMFiles	FM 6.0 Only
KbdSelectFMXMLAuthorFiles	FM12 or greater
KbdSelectGeneratedFiles	FM 6.0 Only
KbdSelectNonGeneratedFiles	FM 6.0 Only
KbdSeparations	!vcv - View Menu Command
KbdSession	!fP - Session preferences - File Menu
KbdSetCap	Graphics Palette
KbdSetCap0	Graphics Palette
KbdSetCap1	Graphics Palette
KbdSetCap2	Graphics Palette
KbdSetCap3	Graphics Palette
KbdSetCapOption	Graphics Palette
KbdSetDash	!ldd - set dashed line
KbdSetDash0	Graphics Palette
KbdSetDash1	Graphics Palette
KbdSetDash2	Graphics Palette
KbdSetDash3	Graphics Palette
KbdSetDash4	Graphics Palette
KbdSetDash5	Graphics Palette
KbdSetDash6	Graphics Palette
KbdSetDash7	Graphics Palette
KbdSetDash8	Graphics Palette
KbdSetDashOption	Graphics Palette
KbdSetElCatAll	!ela - List All Elements
KbdSetElCatChild	!elc - List All Elements allowed
KbdSetElCatFreq	!elf - List frequently used Elements
KbdSetElCatLoose	!elu- List valid Elements any order
KbdSetElCatStrict	!elv- List valid Elements working
KbdSetFill	Graphics Palette
KbdSetFill0	Graphics Palette
KbdSetFill1	Graphics Palette
KbdSetFill2	Graphics Palette
KbdSetFill3	Graphics Palette

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdSetFill14	Graphics Palette
KbdSetFill15	Graphics Palette
KbdSetFill16	Graphics Palette
KbdSetFill17	Graphics Palette
KbdSetFill18	Graphics Palette
KbdSetFill19	Graphics Palette
KbdSetFill1A	Graphics Palette
KbdSetFill1B	Graphics Palette
KbdSetFill1C	Graphics Palette
KbdSetFill1D	Graphics Palette
KbdSetFill1E	Graphics Palette
KbdSetFill1F	Graphics Palette
KbdSetFromColor	Graphics Palette
KbdSetKnockOut	Graphics Palette
KbdSetOverPrint	Graphics Palette
KbdSetPen	Graphics Palette
KbdSetPen0	Graphics Palette
KbdSetPen1	Graphics Palette
KbdSetPen2	Graphics Palette
KbdSetPen3	Graphics Palette
KbdSetPen4	Graphics Palette
KbdSetPen5	Graphics Palette
KbdSetPen6	Graphics Palette
KbdSetPen7	Graphics Palette
KbdSetPen8	Graphics Palette
KbdSetPen9	Graphics Palette
KbdSetPenA	Graphics Palette
KbdSetPenB	Graphics Palette
KbdSetPenC	Graphics Palette
KbdSetPenD	Graphics Palette
KbdSetPenE	Graphics Palette
KbdSetPenF	Graphics Palette
KbdSetPoster	!fsp set the poster on inset (FM10 or greater)
KbdSetSearch	!fis - Display Set Search dialog

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdSetSep	Graphics Palette
KbdSetSepALL	Graphics Palette
KbdSetSepKeep	Graphics Palette
KbdSetSepResetTIntoVerPrint	Graphics Palette
KbdSetSides	!gn - Graphic Menu Command
KbdSetSolid	!lds- set solid line
KbdSetTint	Graphics Palette
KbdSetWidth	Graphics Palette
KbdSetWidth0	Graphics Palette
KbdSetWidth1	Graphics Palette
KbdSetWidth2	Graphics Palette
KbdSetWidth3	Graphics Palette
KbdSetWidthOption	Graphics Palette
KbdShftSpace	+ [Space] - Shift Space
KbdShowBorders	FM 6.0 Only
KbdShowCondInd	FM 6.0 Only
KbdShowElemBorder	FM 6.0 Only
KbdShowElemCtx	FM 6.0 Only
KbdShowElemCtxAv	FM11 or greater
KbdShowElemTags	FM 6.0 Only
KbdShowFileListPod	FM12 or greater
KbdShowGraphics	FM 6.0 Only
KbdShowGrid	FM 6.0 Only
KbdShowHotSpotIndicators	FM11 or greater
KbdShowlinkBoundaries	FM 6.0 Only
KbdShowRulers	FM 6.0 Only
KbdShowSymbols	FM 6.0 Only
KbdSilentOpen	!oS
KbdSingleQuote	' Smart Quotes
KbdSlow0x00	sets wait-time to 0
KbdSlow0x01	adds 1/100 second
KbdSlow0x10	adds 1/10 second
KbdSlow1x00	adds 1 sec to wait time
KbdSmallToolWin	small tools window

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdSmEqn	!ms - Small equation
KbdSmooth	!gs - Graphic Menu Command
KbdSnap	!gp - Graphic Menu Command
KbdSnap0	Graphics Palette
KbdSnap1	Graphics Palette
KbdSoftHyphen	!-D, ^- - discretionary hyphen
KbdSpelling	!es - Edit Menu
KbdSpellReset	!lr - reset checked pgfs - Spell Checker
KbdSplit	!CS - Split column below IP - Format Menu - Cust. Page Layout
KbdSPOptions	!lo - spell check options - Spell Checker
KbdStuff	~y, !ii - Edit Menu
KbdStripFlowStructure	!ssf - Remove structure from flow.
KbdStrWin	!EV - Structure View.
KbdStuffItem	!df - Set up stuff item.
KbdStyleCatalog	FM11 or greater
KbdStyleDesigner	FM11 or greater
KbdStyleFmtDelete	FM11 or greater
KbdStyleFmtDeleteUnused	FM11 or greater
KbdStyleQuick	FM12 or greater
KbdSubscribeTo	Mac only - Edit Menu
KbdSwapRedBlue	!RedBlue - Swap red and blue for 24-bit frameimage.
KbdSymbols	!vt - View Menu Command
KbdSymFont	temporarily set IP font to Symbol
KbdTab	^i, [Tab] - Tab, just normal tab
KbdTabCenter	MS Windows Only
KbdTabDecimal	MS Windows Only
KbdTableAddRC	!ta - Add Rows or Columns
KbdTableConvert	!tv - Convert to Table/Paragraphs
KbdTableCustRS	!tx - Custom Ruling and Shading
KbdTableExitIP	!tl - Move IP out of table
KbdTableFormat	!td - Table Designer
KbdTabLeft	MS Windows Only
KbdTableIns	!ti - Insert Table
KbdTableResizeCol	!tz - Resize columns

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdTableRowFmt	!tr - Row Format
KbdTableRules	!te - Edit Ruling Style
KbdTableStraddle	!tl - Straddle/Unstraddle
KbdTableTab	![Tab] - Tab for table cells
KbdTableVars	FM 2015 or greater
KbdTableWin	!tcc - table catalog window (FM10 or greater)
KbdTabRight	MS Windows Only
KbdTagstatus	Used by MS Windows
KbdTblDlGAddAbove	!tRa - Add table rows above
KbdTblDlGAddBelow	!tRb - Add table rows below
KbdTblDlGAddLeft	!tcl - Add columns to left
KbdTblDlGAddRight	!tcr - Add columns to right
KbdTblDlGClearEmpty	!tce - Clear leaving cells empty
KbdTblDlGClearX	!tcx - Clear removing cells
KbdTblDlGPasteAfter	!tpa - Paste table after
KbdTblDlGPasteBefore	!tpb - Paste table before
KbdTblDlGPasteRepl	!tpr - Paste replacing selection
KbdTblDlGShrinkWrap	!tw - Shrink wrap column width to
KbdTblDlGUnifyTF	!tut - Unify Table Formats
KbdTblfmtDelete	
KbdTblfmtDeleteUnused	FM10 or greater
KbdTblIPAbove	!tmu - IP to cell above.
KbdTblIPBelow	!tmd IP to cell below.
KbdTblIPBottom	!tmb - IP to bottom cell in current column.
KbdTblIPLeft	!tml - IP to cell on left.
KbdTblIPLeftMost	!tma IP to left most cell in current row.
KbdTblIPNext	!tmn - IP to next logical cell.
KbdTblIPPrev	!tmp - IP to next previous cell.
KbdTblIPRight	!tmr - IP to cell on right.
KbdTblIPRightMost	!tme - IP to right most cell in current row.
KbdTblIPTop	!tmt - IP to top cell in current column.
KbdTblIPTopLeft	!tms - IP to top left cell of table selection.
KbdTblSelCell	!the - Select the current cell, then next.
KbdTblSelCellText	!tha - Select all text in the cell.

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdTblSelCol	!thc - Select the current col, then next.
KbdTblSelColBody	!thb - Select all body cells in the column.
KbdTblSelRow	!thr - Select the current row, then next.
KbdTblSelTable	!tht - Select the current table.
KbdTcSearchBook	FM10 or greater
KbdTcSearchDitamap	FM10 or greater
KbdTcSearchDoc	FM10 or greater
KbdTcSearchSel	FM10 or greater
KbdTcSearchUsername	FM10 or greater
KbdTcToolWin	Removed in FM 9.0
KbdTestModal	!dt - Test document as modal dialog.
KbdTestModeless	!dT - Test document as modeless dialog.
KbdTestPrintDbre	!dp - print dialog resource test
KbdTextColPro	!jpS - Toggle Sideheads On/Off - Other Page Commands
KbdTextLTool	!tl - Tools Window Command
KbdTextOptions	!oto - Text Options - Format Menu - Document
KbdTextRTool	!tr - Tools Window Command
KbdThesaurus	!et - lookup selected word in Thesaurus - Edit Menu
KbdThesaurusReplace	!Tr - replace active selection with word from thesaurus - Edit Menu
KbdThinSpace	![Space]t - thin space = 1/12 em
KbdToc	FM 6.0 Only
KbdToggleCollapse	!Ex - Toggle element collapse
KbdToggleCollapseAll	!EX - Toggle collapse all siblings
KbdToggleCollapseChildren	Er - Toggle collapse all children recursively (FM10 or greater)
KbdToggleInclusion	!eli - Toggle inclusion grouping
KbdToggleDraw	!vv - Toggle draw/don't draw preference - View Menu Command
KbdToolbar	MS Windows Only
KbdToolbarNextPage	MS Windows Only
KbdToolbarPrevPage	MS Windows Only
KbdToolWin	!lw, !gT - tools window
KbdTrackChange	
KbdTrackChangeDisable	FM10 or greater
KbdUIAlertStringsPref	Alert Strings Preference Dialog (FM10 or greater)
KbdUnCond	!qU - Conditional Text "Unconditional" key

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdUndo	!eu, [Undo] - Edit Menu
KbdUngroup	!gu - Graphic Menu Command
KbdUnSmooth	!gm - Graphic Menu Command
KbdUpdateRef	update ref @ ip
KbdUpStack	!vN - Forward Link
KbdUseEltDefsFrom	!fie - Import Element Definitions
KbdUseFmtFrom	!fio - Import Formats - File Menu
KbdValAllow	!vae - Allow as special case
KbdValClear	!vce - Clear special validation cases
KbdValDoc	!vd - Validate document
KbdValElem	!ve - Validate element
KbdValFlow	!vf - Validate flow
KbdValidateHypertext	!vh - Validate hypertext makers. FM 6.0 Only
KbdValidation	!Ev - Validate
KbdValIgnore	!vie - Ignore missing elements
KbdValIgnoreAttr	!via - Ignore missing attributes
KbdValStart	!vn - Start validating
KbdVarCurdate	!ohd - Insert current date # variable - Special Menu Command
KbdVarCurPg	!ohp - Insert current page # variable - Special Menu Command
KbdVariable	!sv - Special Menu Command
KbdVarOther	!oho - Insert current date # variable - Special Menu Command
KbdVarPgCount	!ohc - Insert page count variable - Special Menu Command
KbdVarQuick	!qv, ^0 - Variable quick key
KbdVerifyContext	(unbound) Verify context tables
KbdVideoImport	FM 2017 or greater
KbdViewApi	FM11 or greater
KbdViewApiShortcut	FM11 or greater
KbdViewer	!Flk - Toggle view-only document
KbdViewerMakeDocument	
KbdViewerMakeModelessDialog	
KbdViewerMakePalette	
KbdViewSep1	!v1 - View Menu Command
KbdViewSep2	!v2 - View Menu Command
KbdViewSep3	!v3 - View Menu Command

Table 238: List of F-codes

F-code Identifiers	Description of F-code
KbdViewSep4	!v4 - View Menu Command
KbdViewSep5	!v5 - View Menu Command
KbdViewSep6	!v6 - View Menu Command
KbdViewApiSwitch	FM11 or greater
KbdVScroll	
KbdWaitDlgS	Special for X and FrameServer
KbdWidth0	!0w - Set to thinnest width - Line Width Command
KbdWidth1	!9w - Set to thickest width - Line Width Command
KbdWinArrangeIcon	MS Windows Only
KbdWinCascade	MS Windows Only
KbdWindowFullDown	!vsn - Scroll next screen
KbdWindowFullUp	!vsp - Scroll prev screen
KbdWinTile	MS Windows Only
KbdWrapQuick	!Ew, ^2 - wrap element quick key
KbdXCharS	^t, ~t - transpose characters (eXchange)
KbdYANK	^y !eY - pastes kill buffer - Edit Menu
KbdZoom	zoom
KbdZoom100	!zz - zoom to 100%
KbdZoomAutoFitAfterZoom	!zaf - toggle to enable/disable autofit after zoom
KbdZoomFitPage	!zp - zoom fit page in window
KbdZoomFitTextFrame	!zt - zoom fit window to textframe
KbdZoomFitWindow	!zw - zoom fit window to page
KbdZoomIn	!zi - zoom in
KbdZoomOut	!zo - zoom out
KbdZoomSet	!zs - set zoom percentages
MenuExposeWin	Expose window from menu
MenuHypertext	
MenuImportFile	Import file from menu
MruSelectFile	Select a most recently visited
ObjSelExtendNext	!oe - Extend object selection to - Object Selection
ObjSelFirst	!of - Select first object on cur page - Object Selection
ObjSelLast	Select last object on cur page - Object Selection
ObjSelNext	!on - Select next object on cur page - Object Selection
ObjSelNextWrap	wrap to beginning if at the end - Object Selection

Table 238: List of F-codes

F-code Identifiers	Description of F-code
ObjSelPrev	Select prev object on cur page - Object Selection
ObjSelPrevWrap	wrap to end if at the beginning - Object Selection
OpenMRUFiles	FM12 or greater
PgfAnyplace	!jA - Start anywhere
PgfApplyCatToSel	
PgfApplyTag	Set pgf tag from menu
PgfBam	!jj - Set to pgf dialog values, no dialog
PgfCenter	!jc - Center paragraph
PgfColTop	!jC - Start at top of column
PgfDECLine	!j-, !-1 - Decrement line spacing
PgfDesignCat	
PgfDesignKitApply	
PgfDesignKitNewFormat	
PgfDesignKitUpdateALL	
PgfDesignKitUpdateOptions	
PgfDoubleSpace	!j2 - Double space paragraphs
PgfFull	!jf - Full justify paragraph
PgfHyphenOff	!jn - Turn hyphenation off
PgfHyphenOn	!jh - Turn hyphenation on
PgfIncLine	!j+, !+1 - Increment line spacing
PgfKbdBodyStraddle	!jpt - Pgf placement body straddle
PgfKbdFullStraddle	!jpt - Pgf placement full straddle
PgfKbdRunin	!jpr - Pgf placement run-in
PgfKbdSidebody	!jpn - Pgf placement normal
PgfKbdSideheadFirstBaseline	Align first baseline
PgfKbdSideheadLastBaseline	Align last baseline
PgfKbdSideheadTop	Align Tops
PgfLeft	!jl - Left justify paragraph
PgfLeftTop	!jL - Start at top of left page
PgfLineFix	!jx - Fixed line spacing
PgfLineFloat	!jo - Floating line spacing
PgfLineSpace	!ju - Line Space dialog
PgfNewFormat	!opn - New Pgf Format dialog
PgfOneAndAHalfSpace	!j/ - 1 1/2 space paragraphs

Table 238: List of F-codes

F-code Identifiers	Description of F-code
PgfPageTop	!jP - Start at top of page
PgfRight	!jr - Right justify paragraph
PgfRightTop	!jR - Start at top of right page
PgfSingleSpace	!j1 - Single space paragraphs
PgfSpaceBetween	!jw- Space Between dialog
PgfUnify	!jU - Make all pgfs with current
PgfUpdateAll	!opu - Update Pgf Format dialog
SearchCbOverride	
ShowStructText	FM 2017 or greater
StyleApplyCatToSel	FM11 or greater
SuppressInsets	!eS - Suppress auto reference updating - Text Inset Command
SwCsrDown	
SwCsrLeft	
SwCsrRight	
SwCsrUp	
TblApplyCatToSel	FM10 or greater
TblApplyTag	Set table tag from menu (FM10 or greater)
TblDesignCat	
TblDesignKitApply	
TblDesignKitNewFormat	
TblDesignKitUpdateAll	
TblDesignKitUpdateOptions	
TextInsetProps	!ei - Text inset properties - Text Inset Command
TextSelDragging	Text Selection
TextSelElement	Text Selection
TextSelExtend	Text Selection
TextSelExtendElement	Text Selection
TextSelExtendLine	Text Selection
TextSelExtendPgf	Text Selection
TextSelExtendSent	Text Selection
TextSelExtendWord	Text Selection
TextSelLine	Text Selection
TextSelPgf	Text Selection
TextSelQuickCopy	Text Selection

Table 238: List of F-codes

F-code Identifiers	Description of F-code
TextSelSelectOnly	Text Selection
TextSelSent	Text Selection
TextSelWord	Text Selection
ToggleFluidView	!VF - Toggle fluid view
Txt10	Set text to 10 point
Txt12	Set text to 12 point
Txt14	Set text to 14 point
Txt18	Set text to 18 point
Txt24	Set text to 24 point
Txt7	Set text to 7 point
Txt9	Set text to 9 point
TxtBam	!cc - Set to font dialog values, no dialog
TxtBold	!cb,+[F2] - Set chars to bold
TxtChangeBar	!ch - Set chars to change bar
TxtDblUnderline	!cd - Set chars to double underline
TxtDecSize	!c[, !s - Decrement text size
TxtDefault	!ocp - Set chars to default pgf font
TxtFamilyAndVariationPlatform	Platform specific set text family and variation
TxtFamilyAndVariation	Set text family and variation
TxtFont	Platform specific set text family and variation
TxtIncSize	!c[, !s - Increment text size
TxtInitCaps	~c - Set chars to initial caps
TxtItalic	!ci,+[F3] - Set chars to italic
TxtKern	!ck - Set chars to kerned
TxtLessStretch	![c Condense textstretch by 5 %pts
TxtLowerCase	~l - Set chars to lower case
TxtMiniCaps	!cm - Set chars to small caps
TxtMoreStretch	![e Extend textstretch by 5 %pts
TxtNormal	!c= - Set chars to normal
TxtNormalCase	- Set chars to as-typed case
TxtNoStretch	![n Set textstretch to 100%
TxtNumUnderline	!c2 - Set chars to numeric underline
TxtOtherSize	Set text size with dialog
TxtOutline	Set chars to outline

Table 238: List of F-codes

F-code Identifiers	Description of F-code
TxtOverline	!co - Set chars to overline
TxtPlain	!cp,+[F1] - Set chars to plain
TxtSelBold	- Set chars to bold
TxtSelDecSize	- Decrement text size, sel only.
TxtSelIncSize	- Increment text size, sel only.
TxtSelItalic	- Set chars to italic
TxtSelPlain	- Set chars to plain
TxtSelUnderline	- Set chars to underline
TxtShadow	Set chars to shadow
TxtSpread	![C, !c[Right] - Spread spacing 20% of an em
TxtSqueeze	![D, !c[Left] - Squeeze spacing 20% of an em
TxtStrikeout	!cs,+[F5] - Set chars to strikethrough
TxtStructuredBold	FM 2015 or greater
TxtStructuredItalic	FM 2015 or greater
TxtStructuredUnderline	FM 2015 or greater
TxtSub	!c- - Set chars to subscript
TxtSuper	!c+ - Set chars to superscript
TxtTsume	!ct - Set chars to tsume
TxtUnderline	!cu,+[F4] - Set chars to underline
TxtUpperCase	~u - Set chars to upper case
TxtUSize	Set text to the last size (initially 36)
TypeIn	
UpdateInsets	!eU - Update References - Text Inset Command
WebAdobeBookmarks	FM 6.0 Only
WebCorporateNews	FM 6.0 Only
WebDownloadables	FM 6.0 Only
WebFrameBookmarks	FM 6.0 Only
WebGoToAdobe	!www - access web services. FM 6.0 Only
WebPreferences	FM 6.0 Only
WebRegistration	FM 6.0 Only
WebTopIssues	FM 6.0 Only

Command Option Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Symbols

!BookMainMenu 85
!MakerMainMenu 85
!QuickMakerMainMenu 85
!ViewOnlyMenu 85

A

Action 23
AddEditTags 6
AddHyperLinks 79
AddTo 84, 85, 86
AFrame 41, 71
AlertUserAboutFailure 35, 57, 113, 133, 147
All 63
AllowInconsistentNumProps 147
AllowNonFMFiles 148
AllowSpecial 92
AllowViewOnlyFiles 148
ANGLE 53
Ascending 142
AttrDefName 93
AttrExpr 5
Attributes 29
AttrName 92
AttrType 93
AutoHyphen 41

B

Backward 38, 40
BeefyDoc 113
Binary 37
BleedBottom 133
BleedInside 133
BleedMarks 133

BleedOutside 133
BleedTop 133
BodyPage 71
BodyRows 97, 98
Book 71
BookComponent 71
BookFileMenu 86
BookIsInUse 113
BookObject 6, 7, 12, 27, 43, 44, 45, 62, 64, 74, 89, 91, 122, 125, 129, 132, 142, 145, 147, 149
Bottom 77
BottomMargin 78
BottomRow 140
Button 15, 16, 17, 19, 75
Button3Label 19
Button4Label 19

C

Cancel 130
CAPITALIZATION 53
Capitalization 42
Case 40, 142
Cell 71
CellObject 143
CellSeparator 57
Center 39
CHANGEBAR 53
ChangeBar 42
ChangeBars 29
CharFmt 47, 71
CharPropsChange 52
CHARTAG 53
CharTag 41, 42
CheckBox1 19
CheckBox1Label 19

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

CheckBox2 19
CheckBox2Label 19
CheckBox3 19
CheckBox3Label 19
CheckBox4 19
CheckBox4Label 19
CheckDir 15
Child 96
Choice 93
ClosedDocs 146
CmpType 142
COLOR 53
Color 42, 47, 63, 71
ColorBars 133
Column 77
ColumnGap 78
CombinedFontFamily 42
CombinedFonts 63
Command 32, 47, 71
CommandObject 1, 125
CompType 89
Cond 63
CondFmt 5, 8, 47, 71
CondTag 23
CropMarks 133
CutTblCells 11, 14

D

Data 105
Default 93
DeleteCondName 28
DeleteEmptyPages 64
DeleteText 28
Descending 142
Destination 150
Destpage 8
Direction 98, 99

Directory 15
DisallowBookDoc 113
DisallowBookMif 113
DisallowDoc 57, 113
DisallowFilterTypes 57, 114
DisallowGraphicTypes 57
DisallowMacEditions 57
DisallowMif 57, 114
DisallowPlainText 57, 114
DisallowSgml 57, 114
DisallowXml 57, 114
DitaApplyOutputTemples 133
DitaBookChapterTemplate 133
DitaBookChapterTitleTemplate 133
DitaBookIndexTemplate 133
DitaBookLOFTemplate 133
DitaBookLOTTTemplate 133
DitaBookSectionTemplate 133
DitaBookSectionTitleTemplate 133
DitaBookSubsectionTemplate 133
DitaBookSubsectionTitleTemplate 133
DitaBookTitleTemplate 133
DitaBookTOCTemplate 133
DitaCompositeDocTemplate 133
DitaGenerateComponentsAtOneLoc 133
DitaGenerateFlatBook 133
DitaMaxRefLevels 57, 114
DitaPostProcessingOnBook 134
DitaSavePdfViaBook 134
DitavalCondTag 134
DitavalFile 133, 134
Doc 71
DocObject 8, 11, 12, 13, 14, 27, 32, 33, 35,
46, 54, 56, 62, 64, 74, 80, 81, 88, 90, 91, 97,
99, 102, 120, 121, 122, 125, 129, 132, 142,
145, 146, 149
DocType 112
Document 77

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

DocumentProps 63
DoNotImportEDD 64
DoNotImportStructApp 64
DoNotLockFile 57, 114
DontApplyAllRows 120
DontDeleteHiddenText 11, 14, 120
DontNotifyApiClient 35, 57, 114, 134

E

Edit1 19
Edit2 19
Edit3 19
EditMenu 86
Element 27, 41, 122, 145, 150
ElementBegin 52
ElementDef 47, 91, 149
ElementDefName 91, 149
ElementEnd 52
ElementLoc 89, 91
ElementRange 27, 74, 145, 150
ElemPrefixBegin 52
ElemPrefixEnd 52
ElemSuffixBegin 52
ElemSuffixEnd 52
ENCODING 53
EndEvent 30
EndPos 50
EndSub 43, 144
eventname 30
EventProc 84
Everything 146
Expression 28
expression 55, 75, 141
ExprName 44
Extension 112

F

FAMILY 53
FieldName 104
File 23, 30, 34, 35, 56, 65, 71, 109, 110, 111, 113, 120, 124, 132
FileIsGraphic 57
FileIsInUse 114, 134
FileIsMakerDoc 58
FileIsOldVersion 114
FileIsSgmlDoc 58
FileIsStructured 114
FileIsText 58, 114
FileIsXmlDoc 58
FileMenu 86
FileType 135
FilterFormatId 35, 58, 114, 136
FitGraphicInSelectedRect 58
Flag 93
Flow 71
FlowBegin 52
FlowEnd 52
Fn 71
FnAnchor 52
Font 63
FontAngle 42
FontChangedMetric 114
FontFamily 42
FontNotFoundInCatalog 115
FontNotFoundInDoc 115
FontSize 42
FontVariation 42
FontWeight 42
FooterRows 97, 98
Footnote 41
ForceImportAsText 58
ForceOpenAsText 115
ForceUpdate 146

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ForEach 70
Format 80, 97
FormatImportedText 58
FormatMenu 86
Formatted 79, 80
Formatting 146
FormatTypes 62
FrameAnchor 52
FrameClient 9
From 46, 125, 126, 127, 128, 129
FromDocObject 62, 64
FromString 50
FromTextLoc 40

G

GraphicDpi 35, 58
GraphicsFmt 47
GraphicType 83

H

HeaderRows 97, 98
Height 78
HowToImport 58
Hyperlinks 29
Hyphenating 146

I

IFF 53
IgnoreMods 12
ImportAsType 58
ImportFlags 64
ImportFlowPageSpace 58
ImportFlowTag 59
ImportTblTag 59
In 71, 129

InCond 41
Incr 70
Indirect 37, 142
Init 15, 16, 17
InitVal 70
InlineComponentType 79
InList 37
InObject 39, 51
InRange 39, 51
InsertBelowRight 120
InsertCondName 28
InsetElementDef 59
InString 38
Interactive 11, 13, 14, 43, 120
Internal 146
IntoFirst 74
IntoLast 74
IntValue 102
Invisible 78
IOType 111, 120
IsErrorDoc 115

K

KERNX 53
KERNY 53

L

Label 65, 84, 85
Label1 19
Label2 19
Label3 19
LANGUAGE 53
Language 42
LanguageNotAvailable 115
LeaveHeadingRowsEmpty 59
Left 77

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

LeftCol 140
LeftInsideMargin 78
LineBegin 52
LineEnd 52
List 17, 142
listoftextitemtypes 51
LockCantBeReset 115, 136
LoopVar 70, 71
LowerCase 50

M

MakeIconic 115
MakePageCount 136
MakeVisible 112, 115, 148
ManualUpdate 59
MappingFile 59
Marker 41, 71
MarkerAnchor 52
MarkerName 79
MarkerText 41
MarkerType 47, 48
MarksOffset 136
MarksWeight 136
MasterPage 47, 71
Math 63
Member 2, 37, 126, 129
Menu 47, 71
MenuItemSeparator 47
MenuObject 3, 127
MenuSepObject 4, 127
Message 9
ModDateChanged 137
Mode 15, 75

N

Name 46, 65, 81, 84, 85, 86, 87, 89, 91, 144
Named 87
NameStripe 115
NewDocObject 28
NewVar 15, 16, 17, 44, 45, 46, 47, 50, 51, 55, 56, 76, 77, 78, 80, 81, 82, 83, 84, 85, 86, 88, 89, 90, 91, 92, 93, 96, 97, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 120, 124, 142
NoCase 38, 39
NoNotify 64
NoStructuredErrorLog 137
NotInCond 41
NoWrap 40
Number 46, 77, 126, 129
NumCellSeparators 59
NumCols 24, 78, 97, 99, 143
NumColumns 59
NumRows 25, 143

O

Object 13, 24, 99, 103
ObjectStyleTag 41
Offset 96, 103, 104
OldDocObject 28
OpenAsType 115
OpenComposite 29
OpenDocs 146
OpenDocViewOnly 115
OpenFileNotWritable 115
OpenId 115
OpenOldVersionFiles 148
openoptions 76, 112, 113
OpenSummary 29
OUTLINE 53

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

OVERLINE 53

Overline 42

P

Page 63

PageBegin 52

PageEnd 52

PageHeight 136

PageInfo 136

PageWidth 136

PAIRKERN 53

Parent 96

ParentObject 81, 82, 83

Parm 131

Path 108

PDFColorSpace 136

PDFCompatibility 136

PDFEmbedPageThumbnails 136

PDFLayout 136

PDFOpenPage 136

PDFOptimizedWebView 136

PDFPageNum 59

PDFPageRangeEnd 136

PDFPageRangeStart 136

PDFPages 136

PDFPassword 136

PDFPreset 136

PDFPrimaryOutput 136

PDFStandard 136

PDFTag 136

PDFUseDistiller 136

PDFView 136

PDFZoom 137

Pgf 63, 72

PgfBegin 52

PgfEnd 52

PgfFmt 47, 48, 72

PgfFmtName 79, 90

PgfTag 41

POSITION 53

Position 42

Predetermined 77

Prefix 38, 39

PrevObject 88, 89, 90

Properties 8

Property 128

property 141

PropertyName 4, 107

PropertyValue 4, 108

Pub 41

PublicId 112

PublishLogFilePath 137

R

RangeMax 93

RangeMin 93

RasterDpi 59

RasterImageHeight 59

RasterImageWidth 59

ReDisplay 146

RefFileNotFound 116, 137

RefPage 47, 63, 72

RegexFlavor 41

RegMarks 137

Remove 5

RemoveBookInfo 64

RemoveChars 50

RemoveExceptions 63

RemoveLeading 50

RemoveManualPageBreaks 59

RemoveOverrides 59, 64

RemovePageBreaks 63

RemoveTrailing 50

ReplaceCells 120

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ReplaceWithNull 9
Required 93
ResetEquationSettings 146
ResetRefFrames 146
RestartPgfnNums 146
RetainNameStripe 137
ReturnComposite 29
ReturnElementRange 41
ReturnFilename 35, 56, 113, 132
ReturnIndex 17
ReturnLength 38
ReturnNativeError 35, 56, 113, 132
ReturnPos 37, 38
ReturnRange 39, 41
returns 131
ReturnStatus 35, 37, 38, 39, 41, 56, 112, 113, 132, 147
ReturnString 38, 39
ReturnSummary 29
ReturnVal 9
Reverse 50
Rewind 124
Right 77
RightCol 140
RightOutsideMargin 78
Row 72
RowObject 25, 98
Rubi 42
RubiCompositeBegin 52
RubiCompositeEnd 52
RubiTextBegin 52
RubiTextEnd 52
RuleType 82
RulingFmt 47, 72

S

SaveFileNotWritable 137
saveoptions 132
SaveTextExtraBlankLineAtEOP 137
SaveTextTblSetting 137
ScriptName 9
ScrollTo 39, 40
searchobject 40
SelectTable 140
Settings 7
SgmlImportApplication 59
SgmlOpenApplication 116
SgmlSaveApplication 137
SHADOW 53
Shortcut 65, 84
ShowBookErrorLog 148
ShowBrowser 35, 59, 116
ShowRasterDpiDialog 59
ShowSaveTextDialog 137
SidedNess 77, 78
SIZE 53
SkipStep 130, 131
SlugArea 138
Sourcepage 8
SPREAD 53
Spread 42
Start 38, 39
StartCol 24, 99
StartPos 50
STRETCH 53
Stretch 42
STRIKETHROUGH 53
StrikeThrough 42
String 38, 39, 41, 52
stringexpression 99, 150
StructuredApplication 112
StructuredImportApplication 59

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

StructuredOpenApplication 116
StructuredSaveApplication 138
SubCol 72
SubColBegin 52
SubColEnd 52
Subname 110
subname 43, 144
Suffix 38, 39
SummaryOnly 29
SystemId 112

T

Tab 107
TabCenter 107
TabDecimal 107
Table 41, 63
TabLeader 107
TabLeft 107
TableObject 24, 25, 98, 99, 140
TabRelativeCenter 107
TabRelativeDecimal 107
TabRelativeLeft 107
TabRelativeRight 107
TabRight 107
TabType 107
Tbl 72
TblAnchor 52
TblFmt 47, 72
TblNumHeadingRows 60
Template 76
TemplateShouldInsertRoot 116
Text 90
TextFrame 72
TextFrameBegin 52
TextFrameEnd 52
TextInset 41, 146
TextInsetBegin 52

TextInsetContents 26
TextInsetEnd 52
TextInsetName 60
TextLoc 55, 56, 79, 80, 91, 97, 99
TextObjId 53
TextRange 8, 40
textrangevar 25
Threshold 29
Title 15, 16, 17, 19
Title2 19
To 1, 2, 3, 4
Top 77
TopMargin 78
TopRow 140
TreatParaAsRow 60
TSUME 53
Type 15, 16, 17, 19, 47, 143

U

UnanchoredFrame 47, 48
UndefAttr 125, 129
UNDERLINING 53
Underlining 42
Units 16
UNIXpermissions 138
UnresolvedTextInset 41
UnresolvedXRef 41
Until 69
Update 146
UpdateBookGeneratedFiles 148
UpdateBookInlineComponents 148
UpdateBookMasterPages 148
UpdateBookNumbering 148
UpdateBookOLELinks 148
UpdateBookTextReferences 148
UpdateBookXRefs 148
UpdateBrowserDirectory 116

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

UpdateFRVList 138

UpdateMToc 116

updateOptions 147

UpdateTextReferences 116

UpdateXRefs 116

UpperCase 50

UseAutoSaveFile 116

UseDefaultUNIXpermissions 138

UseHTTP 60

UseMainFlow 60

UseRecoverFile 116

UseRegEx 40

V

Value 92, 100, 101, 102, 105, 130

Var 26, 63, 72

VarBegin 53

VarEnd 53

VarFmt 47, 72

Variable 41

Variables 146

VARIATION 54

varname 55, 141

varname.property 141

VerifyInsets 116

Visible 43, 77

VisibleOnly 11, 13, 14, 120

W

WEIGHT 54

When 65

While 69

WholeWord 38, 39, 40

Width 78

Wildcard 40

With 129

X

X 106

XRef 41, 63, 72, 146

XRefBegin 53

XRefEnd 53

XRefFmt 47, 72

XRefs 146

Y

Y 106

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Property Name Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

AcrobatBookmarkDisplayTags 154, 171
AcrobatLevel 253
ActiveBook 258
ActiveDoc 258
AddMarkerTypeToStandardMarkers 258
AdjHyphens 248, 253
AFrameIsCropped 201, 208
AFrameIsFloating 201, 208
Alignment 201, 208
AllowAsSpecialCase 264
AllowBaseFamilyBoldedAndObliqued 169
AlsoInsert 270
AnchorInline 203
AnchorRunIntoParagraph 204
AnchorSubcolFarthest 204
AnchorTop 203
AnchorType 201, 209
Angle 201, 203, 204, 205, 206, 207, 209, 212, 213, 214, 215, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 234, 235, 236, 237, 239
ApplyFormatRules 258, 259, 260, 261, 262
ArrowBaseAngle 201, 204, 206, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232, 235, 237
ArrowLength 201, 204, 206, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232, 235, 237
ArrowScaleFactor 201, 204, 206, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232, 235, 238
ArrowScaleHead 201, 204, 206, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232, 235, 238
ArrowTipAngle 201, 204, 206, 213, 215,

218, 220, 222, 224, 226, 228, 230, 232, 235, 238
ArrowType 202, 204, 206, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232, 235, 238
AttrDisplay 264
AttributeDefs 270
Attributes 264
AttributeValueInvalid 264
AutoBackup 258
AutoChangeBars 171
AutoNumChar 248, 253
AutoNumString 248, 253
AutoSave 258
AutoSaveSeconds 258

B

BaseFamily 169
BaselineOffset 202, 209
BasePointX 220, 235
BasePointY 220, 235
Black 167
BlockLines 248, 253
BookComponent(Read-Only) 265
BookComponentIsGeneratable(Read-Only) 160, 161
BookComponentMissing(Read-Only) 265
BookComponentParent 160
BookComponentTemplatePath 160
BookComponentTitle 160
BookComponentType 160
BookDontUpdateReferences 154
BookIsModified(Read-Only) 154
BookIsSelected 154
BookIsViewOnly 154

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

BookParent(Read-Only) 160
BorderWidth 202, 204, 206, 209, 213, 215,
218, 220, 222, 224, 227, 229, 230, 232, 235,
238
BottomMargin 172
BottomRowSelection(Read-Only) 280
BottomSeparator 248, 253

C

CanHaveCheckMark 243
Capitalization 165, 172, 249, 253, 291
CellAboveInCol(Read-Only) 276
CellAngle 276
CellBelowInCol(Read-Only) 276
CellBottomMargin 249, 253
CellColNum(Read-Only) 276
CellDefaultBottomRuling(Read-Only) 276
CellDefaultLeftRuling(Read-Only) 276
CellDefaultRightRuling(Read-Only) 276
CellDefaultTopRuling(Read-Only) 276
CellIsShown(Read-Only) 276
CellIsStraddled(Read-Only) 276
CellLeftMargin 249, 253
CellMarginsFixed 249, 254
CellNumColsStraddled(Read-Only) 276
CellNumRowsStraddled(Read-Only) 277
CellOverrideBottomRuling 277
CellOverrideFill 277
CellOverrideLeftRuling 277
CellOverrideRightRuling 277
CellOverrideShading 277
CellOverrideTopRuling 277
CellRightMargin 249, 254
CellRow(Read-Only) 277
CellTopMargin 249, 254
CellUseOverrideBRuling 277
CellUseOverrideFill 277
CellUseOverrideLRuling 277
CellUseOverrideRRuling 277
CellUseOverrideShading 277
CellUseOverrideTRuling 277
CellVAlignment 249, 254
ChangeBar 165, 166, 172, 173, 191, 199,
209, 236, 249, 254, 265, 272, 280, 284, 291,
292
ChangeBarColor 172
ChangeBarDistance 172
ChangeBarPosition 172
ChangeBarThickness 172
ChapNumComputeMethod 160, 172
ChapterNumber 160, 172
ChapterNumStyle 161, 172
ChapterNumText 161, 172
CharTag 165, 173, 291
CheckMarkIsOn 243
CmdAlert 197
CmdAlertTitle 197
CmdError 197
CmdExit 197
CmdGoToLink 197
CmdGoToLinkFitWin 197
CmdGoToNew 197
CmdGoToObjectId 197
CmdGoToObjectIdFitWin 197
CmdGoToPage 197
CmdMatrix 197
CmdMessage 197
CmdNewLink 197
CmdNextPage 197
CmdNoop 197
CmdOpenLink 197
CmdOpenLinkFitWin 197
CmdOpenNew 197
CmdOpenObjectId 197

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

CmdOpenObjectIdFitWin 197
CmdOpenPage 197
CmdPopup 197
CmdPreviousLink 197
CmdPreviousLinkFitWin 197
CmdPreviousPage 197
CmdQuit 197
CmdQuitAll 197
CmdUnknown 197
ColGap 173
ColGapWidth 209, 232
Color 165, 173, 202, 204, 206, 209, 213, 215, 218, 220, 222, 224, 227, 229, 230, 232, 235, 238, 249, 251, 254, 256, 291
ColorOverprint 167
ColorPrintCtl 168
ColorTintPercent 167
ColorViewCtl 168
ColumnsAreBalanced 209, 232
CombinedFontFamily 165, 173, 249, 254, 291
Comment 270, 271
ComponentDisplayText 161
ComponentElement 161
ComponentIsSelected 161
CondFmtIsShown 169, 173, 174, 278, 279
ContentHeight(Read-Only) 200, 264, 275, 277, 280
ContentIsLooselyValid(Read-Only) 265
ContentIsStrictlyValid(Read-Only) 265
ContentMustBeEmpty(Read-Only) 265
ContentNeededAtBegin(Read-Only) 265
ContextForCSS 274
ContextLabel 274
ContextLabel(Read-Only) 265, 268
CorrespondingXmlPath 161
CountElements 273

CurrentMenuSet 258
CurrentPage 173
CurrentPgf 173
CustomElementList 154, 173
Cyan 168

D

Dash 202, 204, 206, 213, 215, 218, 220, 222, 225, 227, 229, 231, 232, 235, 238
DefaultExclusions 173
DefaultFontAngle 258
DefaultFontFamily 258
DefaultFontVariation 258
DefaultFontWeight 259
DefaultInclusions 173
DestFirstPage 198
DestFluidFlow 198
DestLastPage 198
DestMarker 198
DestMarkerNewLink 198
DestNowhere 198
DestObjectId 198
DestPageNum 198
DestXRef 198
Dictionary 173
Displaying 259
DisplayName(Read-Only) 259
DocAcrobatColumnArticleThreads 154, 173
DocAcrobatDefaultsChanged 154, 173
DocAcrobatElementList 173
DocAcrobatNoArticleThreads 154, 173
DocCharFmtNameList 173
DocCondFmtNameList 174
DocElementDefNameList 174
DocFluidFlow 174

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

DocIsDoubleSided 174
DocIsHelp(Read-Only) 174
DocIsModified(Read-Only) 174
DocIsViewOnly 174
DocMarkerTypeNameList 174
DocMasterPageNameList 174
DocOpenType(Read-Only) 174
DocPgfFmtNameList 174
DocRefPageNameList 174
DocSaveType(Read-Only) 174
DocTblFmtNameList 174
DocVarFmtNameList 175
DocVarFmtNameSystemList 175
DocVarFmtNameUserList 175
DocXRefFmtNameList 174
DontLoadStartupClients 259
DontUpdateTextInsets 175
DontUpdateXRefs 175
DownloadFonts 175
DTheta 204, 209

E

Element 200, 240
Element(Read-Only) 170, 277, 279, 280, 293
Element† 257
ElementBoundaryDisplay 175
ElementCatalog 154, 175
ElementCatalogDisplay 154, 175
ElementDef 265, 273
ElementDefType 271
ElementInCatalog 271
ElementIsCollapsed 265
ElementIsExcludedInContext(Read-Only) 266
ElementIsInvalidInParent(Read-Only) 266

ElementIsInvalidInPosition(Read-Only) 266
ElementIsUndefined(Read-Only) 266
ElementPgfFormat 271
ElementSelection 154, 175
ElementType(Read-Only) 266
ElemPrefixSuffix 274
EnableAlwaysDisable 65, 244
EnableAlwaysEnable 65, 244
EnableBookHasSelection 66, 245
EnableCanPaste 65, 244
EnableCopy 65, 244
EnableCopyFont 65, 244
EnableDocOrBookHasSelection 66, 245
EnabledWhen 243
EnableInCellText 65, 244
EnableInFlow 65, 244
EnableInMath 65, 245
EnableInParaText 65, 245
EnableInTable 65, 245
EnableInTableTitle 65, 245
EnableInText 65, 245
EnableInTextLine 65, 245
EnableIsAFrame 65, 245
EnableIsCell 65, 245
EnableIsCells 66, 245
EnableIsGraphicInset 66, 245
EnableIsObj 66, 245
EnableIsOrInFrame 66, 245
EnableIsTable 245
EnableIsTables 66
EnableIsTextFrame 66, 245
EnableIsTextInset 66, 245
EnableIsTextSel 66, 245
EnableIsViewOnly 66, 245
EnableNeedsBookpOnly 66, 245
EnableNeedsDocpOnly 66, 245
EnableNeedsDocPOrBookP 66, 245
EnableObjProps 66, 245

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

EqnIntegralSizeLarge 175
EqnIntegralSizeMed 175
EqnIntegralSizeSmall 175
EqnLevel1SizeLarge 175
EqnLevel1SizeMed 175
EqnLevel1SizeSmall 175
EqnLevel2SizeLarge 175
EqnLevel2SizeMed 175
EqnLevel2SizeSmall 175
EqnLevel3SizeLarge 175
EqnLevel3SizeMed 175
EqnLevel3SizeSmall 175
EqnSigmaSizeLarge 175
EqnSigmaSizeMed 175
EqnSigmaSizeSmall 176
ErrorInBookComponent(Read-Only) 266
Exclusions 271
ExpandOMaticParent(Read-Only) 243
ExportFilters(Read-Only) 259
ExtractElementTags 161
ExtractTags 161

F

FamilyName 168
Fcode(Read-Only) 242, 243, 244
Fcodes(Read-Only) 243
FileExtensionOverride 176
FileInfoPacket 154, 176
Fill 202, 205, 206, 210, 213, 215, 218, 220, 222, 225, 227, 229, 231, 233, 236, 238
FirstAFrame(Read-Only) 233, 275, 280
FirstBodyPageInDoc(Read-Only) 176
FirstCell(Read-Only) 233, 275
FirstCellInRow(Read-Only) 278, 279
FirstCellInTbl(Read-Only) 281

FirstCharFmtInDoc(Read-Only) 176, 181, 189
FirstChildElement(Read-Only) 266
FirstColorInDoc(Read-Only) 176
FirstCombinedFontDefnInDoc(Read-Only) 176
FirstCommandInSession 171, 248, 253, 258, 259, 263, 291
FirstComponentInBookComponent 161
FirstCondFmtInDoc(Read-Only) 176
FirstElementDefInDoc 154
FirstElementDefInDoc(Read-Only) 176
FirstFlowInDoc(Read-Only) 176
FirstFmtChangeListInDoc 154
FirstFmtChangeListInDoc(Read-Only) 176
FirstFn(Read-Only) 233, 275, 281
FirstFnInDoc(Read-Only) 176
FirstGraphicInDoc(Read-Only) 176
FirstGraphicInFrame(Read-Only) 202, 238
FirstGraphicInGroup(Read-Only) 214
FirstIndent 249, 254
FirstMarkerInDoc(Read-Only) 177
FirstMarkerTypeInDoc(Read-Only) 177
FirstMasterPageInDoc(Read-Only) 177
FirstMenuItemInMenu 242
FirstMenuItemInSession(Read-Only) 259
FirstOpenBook(Read-Only) 259
FirstOpenDoc(Read-Only) 259
FirstPageNum 161, 177
FirstPageVerso 177
FirstPgf(Read-Only) 200, 233, 275, 277, 281
FirstPgfFmtInDoc(Read-Only) 177
FirstPgfInDoc(Read-Only) 177
FirstPgfInFlow(Read-Only) 199
FirstPgfInMainFlow(Read-Only) 177, 181
FirstPgfRules† 271
FirstRefPageInDoc(Read-Only) 177

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- FirstRowInTbl(Read-Only) 281
- FirstRubiInDoc(Read-Only) 177
- FirstRulingFmtInDoc(Read-Only) 177
- FirstSelectedCellInTbl(Read-Only) 281
- FirstSelectedComponentInBook 154
- FirstSelectedGraphicInDoc(Read-Only) 177
- FirstSelectedTiInDoc(Read-Only) 177
- FirstSubCol(Read-Only) 233
- FirstTblFmtInDoc(Read-Only) 177
- FirstTblInDoc(Read-Only) 177
- FirstTextFrameInFlow(Read-Only) 199
- FirstTiInDoc(Read-Only) 177
- FirstVarFmtInDoc(Read-Only) 177
- FirstVarInDoc(Read-Only) 177
- FirstXRefFmtInDoc(Read-Only) 177
- FirstXRefInDoc(Read-Only) 177
- Flow(Read-Only) 233
- FlowIsAutoConnect 199, 210
- FlowIsFeathered 199
- FlowIsPostScript 199, 210
- FlowIsSynchronized 199
- FMBinDir(Read-Only) 259
- FMCurrentDir(Read-Only) 259
- FMHelpDir(Read-Only) 259
- FMHomeDir(Read-Only) 259
- FMInitDir(Read-Only) 259
- FMSgmlDir 259
- FMStructureDir 259
- Fmt 171, 293
- FmtChangeList† 274
- FmtChangeListInCatalog 272
- FmtChangeListTag 274
- FmtRule† 274
- FmtRuleClause 273
- FmtRuleClauses† 273
- FmtRuleType 273
- FMXmlDir 259
- FnCustNumString 161, 177
- FnFirstNum 161, 177
- FnFmt 177
- FnHeightPerCol 177
- FnInstancePosition 178
- FnInstancePrefix 178
- FnInstanceSuffix 178
- FnNum(Read-Only) 200
- FnNumberingPerPage 178
- FnNumComputeMethod 162, 178
- FnNumStyle 162, 178
- FnRefPosition 178
- FnRefPrefix 178
- FnRefSuffix 178
- FOCONTAINER 270
- FOEQN 270
- FOFOOTNOTE 270
- FOGRAPHIC 270
- FOMARKER 270
- FontAngle 165, 179, 249, 254, 291
- FontAngleNames(Read-Only) 260
- FontEncodingName(Read-Only) 165, 169, 179, 249, 254
- FontFamily 165, 179, 249, 254, 291
- FontFamilyAttributes(Read-Only) 260
- FontFamilyNames(Read-Only) 260
- FontPlatformName 165, 179, 249, 254, 291
- FontPostScriptName 165, 179, 250, 254, 291
- FontSize 165, 179, 250, 254, 291
- FontVariation 165, 179, 250, 254, 291
- FontVariationNames(Read-Only) 260
- FontWeight 165, 179, 250, 254, 291
- FontWeightNames(Read-Only) 260
- FormatOverride 179, 250, 266
- FormatTag 274
- FormattingLocked 266
- FOSYSVAR 270
- FOTBL 270

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

FOTBLBODY 270
FOTBLCELL 270
FOTBLFOOTING 270
FOTBLHEADING 270
FOTBLROW 270
FOTBLTITLE 270
FOXREF 270
FrameParent 202, 205, 207, 214, 215, 218,
220, 222, 225, 227, 229, 231, 233, 236, 238
FrameParent(Read-Only) 275
Functions 179

G

GeneralRule 271
GeneralRuleErrorOffsets(Read-Only) 271
GenerateAcrobatInfo 155, 179
GenerateInclude 162
GraphicCantBeSelected 202, 205, 207, 214,
215, 218, 220, 222, 225, 227, 229, 231, 233,
236, 238
GraphicIsButton 233
GraphicIsSelected 202, 205, 207, 214, 215,
218, 220, 222, 225, 227, 229, 231, 233, 236,
238
Gravity 260
GreekSize 260
GroupParent 205, 207, 214, 216, 218, 220,
222, 225, 227, 229, 231, 233, 236

H

HasShiftOrUnshiftCommand 243
HeadArrow 202, 205, 207, 214, 216, 218,
220, 223, 225, 227, 229, 231, 233, 236, 238

Height 202, 203, 205, 206, 207, 208, 210,
211, 214, 216, 217, 218, 219, 220, 221, 223,
224, 225, 226, 227, 228, 229, 230, 231, 232,
233, 235, 236, 237, 238, 239, 276
Height(Read-Only) 275, 279, 291
HelpLink 243
HiddenPage(Read-Only) 179
HighestLevelElement 155, 199
HoleBeforeElement(Read-Only) 266
HostName(Read-Only) 260
HypertextBadMatrixSize 195
HypertextBadSyntaxPathSpec 195
HypertextCantOpenDestFile 195
HypertextCommandIllegalWithinPopup 196
HypertextCommandText 179
HypertextDestinationLinkNotFound 196
HypertextDoValidate 179
HypertextEmptyCommand 195
HypertextExpectedANumberParam 195
HypertextExtraArguments 195
HypertextFcodeInvalid 196
HypertextFileNotMakerDoc 196
HypertextFileNotRegular 196
HypertextFlowMissingLines 196
HypertextHelpDirNotFound 195
HypertextMatrixCommandInvalid 196
HypertextMissingArguments 195
HypertextMissingPopupLabelItem 196
HypertextMissingPopupMarker 196
HypertextNoNamedFlow 196
HypertextObjectIDNotFound 196
HypertextPageNameNotFound 196
HypertextParseBadParam(Read-Only) 179
HypertextParsedArgs(Read-Only) 179
HypertextParsedClientName(Read-Only)
179
HypertextParsedCmdCode(Read-Only) 180

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

HypertextParsedCmdDest(Read-Only) 180
HypertextParsedCmdDestObjID(Read-Only) 180
HypertextParsedCmdDestObjType(Read-Only) 180
HypertextParsedCmdMatrixColumns(Read-Only) 180
HypertextParsedCmdMatrixRows(Read-Only) 180
HypertextParsedDIFFileName(Read-Only) 180
HypertextParsedFlowName(Read-Only) 180
HypertextParsedLinkName(Read-Only) 180
HypertextParsedMessage(Read-Only) 180
HypertextParsedPageName(Read-Only) 180
HypertextParsedTitle(Read-Only) 180
HypertextParseErr(Read-Only) 180
HypertextParseErrMsg(Read-Only) 180
HypertextRecursiveFlow 196
HypertextSyntaxOk 195
HypertextUnanchoredPartialPath 195
HypertextUnrecognizedCommand 195
HypertextUnrecognizedObjectType 196
HypertextUsesDefaultText 196
HypertextValid 196
HypertextValidateErr(Read-Only) 180
Hyphenate 250, 254
HyphMinPrefix 250, 254
HyphMinSuffix 250, 254
HyphMinWord 250, 255

I

IconBarOn 260
ImportFilters(Read-Only) 260
ImportFmtInclude 162

Inclusions 271
InCond 180, 279, 291
InitStructurePattern 271
InkName 168
InsertLinks 162
InsetDpi 210, 216
InsetEditor 216
InsetFile 216
InsetIsFixedSize 216
InsetIsFlippedSideways 216
InsetLinkToText property (*GraphicObject*) **216**
InsetUpdater 216
InTextFrame(Read-Only) 200, 202, 250, 277, 291
InTextObj(Read-Only) 200, 202, 250, 277, 291
InvalidHighestLevel(Read-Only) 266
InvariantName 241
IsCellSelected(Read-Only) 277
IsIconified 155, 180, 259, 260
IsInFront 155, 180, 260
IsOnScreen 155, 180, 260, 261, 263
IsTextRange 274

K

KeepWithNext 250, 255
KeepWithPrev 250, 255
KernX 166, 180, 250, 255, 291
KernY 166, 181, 250, 255, 291
KeyboardShortcutLabel 243
KeyboardShortcuts 240, 244

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

L

Label 153, 155, 181, 242, 244, 246, 260
Labels 244
Language 166, 250, 255
Language(Read-Only) 261
LastAFrame(Read-Only) 233, 275, 281
LastBodyPageInDoc(Read-Only) 181
LastCell(Read-Only) 233, 275
LastCellInRow(Read-Only) 279
LastCellInTbl(Read-Only) 281
LastChildElement(Read-Only) 266
LastFn(Read-Only) 233, 275, 281
LastGraphicInFrame(Read-Only) 202, 238
LastGraphicInGroup(Read-Only) 214
LastMasterPageInDoc(Read-Only) 181
LastPgf(Read-Only) 200, 233, 275, 277, 281
LastPgfRules(Read-Only) 271
LastRefPageInDoc(Read-Only) 181
LastRowInTbl(Read-Only) 281
LastSelectedCellInTbl(Read-Only) 281
LastSubCol(Read-Only) 233
LastTextFrameInFlow(Read-Only) 199
LastUpdate 287, 288, 289, 290
Leading 250, 255
LeftColNum(Read-Only) 281
LeftIndent 250, 255
LeftMargin 181
LeftMasterPage(Read-Only) 181
LetterSpace 250, 255
LineAscent(Read-Only) 292
LineBaseLine(Read-Only) 292
LineBreakAfter 181
LineCap 202, 205, 207, 214, 216, 218, 220,
223, 225, 227, 229, 231, 234, 236, 238
LineDescent(Read-Only) 292
LineSpacing 251, 255
Locked 170, 251, 281, 292, 293

LocX 202, 205, 207, 210, 214, 217, 218,
220, 223, 225, 227, 229, 231, 234, 236, 238
LocX(Read-Only) 275, 279, 292
LocY 202, 205, 207, 210, 214, 217, 218,
221, 223, 225, 227, 229, 231, 234, 236, 238,
292
LocY(Read-Only) 275, 279

M

Magenta 168
MainFlowInDoc(Read-Only) 181
MarkerNames(Read-Only) 261
MarkerText 240
MarkerType 240
MarkerTypeId 240
MarkerTypeNames(Read-Only) 181
MasterPage 246
MatchesContextInUserString (Read-Only)
266
MatchingFirstPgfClauses(Read-Only) 267
MatchingLastPgfClauses(Read-Only) 267
MatchingObjectClauses(Read-Only) 267
MatchingPrefixClauses(Read-Only) 267
MatchingSuffixClauses(Read-Only) 267
MatchingTextClauses(Read-Only) 267
MathFullForm 221
MathSize 210, 221
MaxBottomMargin 181
MaxFirstIndent 181
MaxFontSize 181
MaxInterlinePadding 199
MaxInterPgfPadding 199
MaxJLetSpace 251, 255
MaxJRomSpace 251, 255
MaxLeading 181

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

MaxLeftIndent 181
 MaxLeftMargin 181
 MaxRightIndent 181
 MaxRightMargin 181
 MaxSpace 251, 255
 MaxSpaceAbove 181
 MaxSpaceBelow 181
 MaxSpread 181
 MaxStretch 181
 MaxTabPosition 182
 MaxTopMargin 182
 MenuBar 182
 MenuItemIsEnabled(Read-Only) 242, 243, 245
 MenuItemType(Read-Only) 243
 MenuType(Read-Only) 242
 MinBottomMargin 182
 MinFirstIndent 182
 MinFontSize 182
 MinHang 199
 MinJLetSpace 251, 255
 MinJRomSpace 251, 255
 MinLeading 182
 MinLeftIndent 182
 MinLeftMargin 182
 MinRightIndent 182
 MinRightMargin 182
 MinSpace 251, 255
 MinSpaceAbove 182
 MinSpaceBelow 182
 MinSpread 182
 MinStretch 182
 MinTabPosition 182
 MinTopMargin 182
 Mode(Read-Only) 243

N

n 217
 Name 155, 162, 166, 168, 169, 171, 199, 238, 241, 247, 248, 251, 255, 272, 284, 286, 287, 288, 289, 290, 293
 Name(Read-Only) 182, 242, 243, 245
 Name† 271
 NameSpace 267
 NameSpaceScope(Read-Only) 267
 NarrowRubiSpaceForJapanese 182
 NarrowRubiSpaceForOther 182
 NewElemAttrDisplay 155, 182
 NewElemAttrEditing 155, 182
 NextAFrame(Read-Only) 203
 NextCell 277
 NextCellInRow(Read-Only) 277
 NextCellInTbl(Read-Only) 278
 NextCharFmtInDoc(Read-Only) 166
 NextColorInDoc(Read-Only) 168
 NextCombinedFontDefnInDoc(Read-Only) 169
 NextCommandInSession(Read-Only) 244
 NextComponentInBook 162
 NextCondFmtInDoc(Read-Only) 169
 NextElementDefInDoc(Read-Only) 271
 NextFlowInDoc(Read-Only) 199
 NextFmtChangeListInDoc 272
 NextFn(Read-Only) 200
 NextFnInDoc(Read-Only) 200
 NextGraphicInDoc(Read-Only) 203, 205, 207, 210, 214, 217, 219, 221, 223, 225, 227, 229, 231, 234, 236, 238
 NextGraphicInFrame 205, 207, 214, 217, 219, 221, 223, 225, 227, 229, 231, 234, 236, 238
 NextGraphicInGroup(Read-Only) 205, 207, 214, 217, 219, 221, 223, 225, 227, 229, 231, 234, 236, 239

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

NextInvalidElement(Read-Only) 267
NextMarkerInDoc(Read-Only) 176, 240, 241
NextMarkerTypeInDoc 241
NextMenuItemInMenu 242, 244, 246
NextMenuItemInSession(Read-Only) 242, 244, 245
NextOpenBookInSession(Read-Only) 155
NextOpenDocInSession(Read-Only) 182
NextPgFmtInDoc(Read-Only) 255
NextPgInDoc(Read-Only) 251
NextPgInFlow(Read-Only) 251
NextRowInTbl(Read-Only) 279
NextRubiInDoc(Read-Only) 258
NextRulingFmtInDoc(Read-Only) 286
NextSelectedCellInTbl(Read-Only) 278
NextSelectedComponentInBook(Read-Only) 162
NextSelectedGraphicInDoc 217
NextSelectedGraphicInDoc(Read-Only) 203, 205, 207, 214, 219, 221, 223, 225, 227, 229, 231, 234, 236, 239
NextSiblingElement(Read-Only) 267
NextSubCol(Read-Only) 275
NextTag 251, 255
NextTblFmtInDoc(Read-Only) 284
NextTblInDoc(Read-Only) 281
NextTextFrameInFlow 234
NextTiInDoc(Read-Only) 287, 288, 289, 290
NextVarFmtInDoc(Read-Only) 294
NextVarInDoc(Read-Only) 293
NextXRefFmtInDoc(Read-Only) 171
NextXRefInDoc(Read-Only) 170
NumAtEnd 251, 255
Numbers 182
NumCols(Read-Only) 182

NumColumns 210, 234
NumPoints 225, 227
NumPoints(Read-Only) 219
NumTabs(Read-Only) 251, 255

O

Object(Read-Only) 268
ObjectDataLink 198
ObjectFmtRules(Read-Only) 271
ObjectGraphic 198
ObjectMarker 198
ObjectPgf 198
ObjectTextInset 198
ObjectType 271
ObjectUnknown 198
ObjectXref 198
OldTypeNum 241
OpenDir 261
OperatingSystem(Read-Only) 261
OptJLetSpace 251, 256
OptJRomSpace 251, 256
OptSpace 251, 256
OrphanRows 281, 284
Outline 166, 182, 251, 256, 292
Overflowed 278
Overflowed(Read-Only) 200, 275, 281
Overline 166, 182, 251, 256, 292
Overprint 203, 205, 207, 210, 214, 217, 219, 221, 223, 225, 227, 229, 231, 234, 237, 239
OyamojiTextRange(Read-Only) 257

P

PageBackground 246

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- PageFrame 247, 248
- PageFrame(Read-Only) 246
- PageFramePage(Read-Only) 239
- PageHeight 183, 247, 248
- PageHeight(Read-Only) 246, 247
- PageIsRecto(Read-Only) 246
- PageNext 247, 248
- PageNext(Read-Only) 246
- PageNum 247, 248
- PageNum(Read-Only) 246
- PageNumbering 162
- PageNumComputeMethod 162, 183
- PageNumString(Read-Only) 246
- PageNumStyle 163, 183
- PagePrefix 163
- PagePrev 247, 248
- PagePrev(Read-Only) 246
- PageRounding 183
- PageSide 163
- PageSuffix 163
- PageWidth 183, 247, 248
- PageWidth(Read-Only) 247
- PairKern 166, 183, 251, 256, 292
- ParentElement(Read-Only) 268
- ParentTextFrame(Read-Only) 275
- Path(Read-Only) 261
- PDFAllNamedDestinations 155, 183, 184
- PDFBookMark 183
- PDFDestsMarked 156, 184
- PDFDocInfo 156
- PDFGenerateForReview 184
- PDFStructure 185
- PDFStructureLevel 251, 256
- PDFViewPDF 185
- Pen 203, 205, 207, 211, 214, 217, 219, 221, 223, 226, 228, 230, 231, 234, 237, 239, 286
- PgfAlignment 251, 256
- PgfCatalogReference 272
- PgfCell 185
- PgfIsAutoNum 251, 256
- PgfNumber(Read-Only) 252
- PgfNumbering 163
- PgfNumComputeMethod 163, 185
- PgfSpellChecked 252
- Placement 252, 256
- Platform(Read-Only) 261
- PointPageNum(Read-Only) 247
- PointPageNumStyle 186
- Points 219
- Points(Read-Only) 226, 228
- PolyIsBezier 226, 228
- Position 166, 186, 252, 256, 292
- PrefixRules(Read-Only) 271
- PrevAFrame(Read-Only) 203
- PrevCell(Read-Only) 278
- PrevCellInRow(Read-Only) 278
- PrevCellInTbl(Read-Only) 278
- PrevComponentInBook 163
- PrevFn(Read-Only) 200
- PrevGraphicInFrame 205, 207, 214, 219, 221, 224, 226, 228, 230, 232, 234, 237, 239
- PrevGraphicInGroup(Read-Only) 206, 207, 214, 217, 219, 221, 224, 226, 228, 230, 232, 234, 237, 239
- PrevMenuItemInMenu 242, 244, 246
- PrevPgfInFlow(Read-Only) 252
- PrevRowInTbl(Read-Only) 279
- PrevSelectedCellInTbl(Read-Only) 278
- PrevSiblingElement(Read-Only) 268
- PrevSubCol(Read-Only) 275
- PrevTextFrameInFlow 234
- PrintBlankPages 157, 186
- PrintCollated 157, 186
- PrintCols 186
- PrintDownloadAsianFonts 157
- PrintDownloadTrueTypeAsType1 157

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

PrintEmulsion 157, 186
PrintEndPage 186
PrintEndPoint 187
PrinterName 157, 186, 187
PrintEvenPages 157, 187
PrintFileName 157, 187
PrintImaging 157, 187
PrintInclude 163
PrintLastSheetFirst 157, 187
PrintLowRes 157, 187
PrintManualFeed 187
PrintNumCopies 157, 187
PrintOddPages 157, 187
PrintPaperHeight 157, 187
PrintPaperWidth 157, 187
PrintRegistrationMarks 157, 187
PrintRows 187
PrintScale 157, 187
PrintScope 187
PrintSeps 157, 187
PrintStartPage 186, 187, 188, 292
PrintStartPoint 187
PrintThumbnails 187
PrintToFile 157, 187
PrintTomboMarks 157
ProcessNumber 262
ProductIsDemo 262
ProductIsStructured 262
ProductName(Read-Only) 262
Public 241

R

Radius 211, 232
RectangleIsSmoothed 230
RectangleIsSmoothed(Read-Only) 207

Reformatting 262
RememberMissingFontNames 262
Required(Read-Only) 241
ReservedColor 168
RetainUndoState 262
RightColNum(Read-Only) 282
RightIndent 252, 256
RightMargin 188
RightMasterPage(Read-Only) 188
RowHasSelection(Read-Only) 279
RowIsShown(Read-Only) 279
RowKeepWithNext 279
RowKeepWithPrev 279
RowMaxHeight 279
RowMinHeight 279
RowStart 279
RowTbl(Read-Only) 280
RowType(Read-Only) 280
RpcProgramNumber 262
RpcPropertyName 262
RubiAlignAtLineBounds 188
RubiElement† 258
RubiFixedSize 188
RubiOverhang 188
RubiSize 188
RubiTextRange 258
RuleClauseType† 274
RulingGap 287
RulingLines 287
RulingPenWidth 287
RulingSep 287
Runaround 206, 208, 211, 215, 217, 219, 221, 224, 226, 228, 230, 232, 234, 237, 239
RunaroundGap 203, 206, 208, 211, 215, 217, 219, 221, 224, 226, 228, 230, 232, 234, 237, 239
RunInSeparator 252, 256

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

S

ScreenHeight 157, 188, 262
ScreenWidth 158, 188, 262
ScreenX 158, 188, 262
ScreenY 158, 188, 262
SelectedTbl(Read-Only) 188
SelectedTblCell(Read-Only) 188
SeparateInclusions 158, 188
SepOverride 169, 188, 280, 292
ServerState 158
ServerUrl 158
SgmlApplication 158, 188
Shadow 166, 188, 252, 256, 292
ShiftOrUnshiftCommand 244
ShowAll 188
ShowCondIndicators 188
ShowElementDescriptiveNames 158
SideHeadGap 211, 234
SideHeadPlacement 211, 235
SideHeadRoomInFlow 199
SideHeadWidth 211, 235
SideOffset 203, 211
SkipBlankSeps 158, 188
SmallCapsSize 188
SmallCapsStretch 188
SmartQuotes 188
SmartSpaces 189
Snap 262
SnapAngle 189
SnapGridUnits 189
SpaceAbove 252, 256
SpaceBelow 252, 256
Spacing 199
Specification 274
SpecificationForCSS 274
SpotColorView 189
Spread 189, 252, 256, 292
StackWarningLevel 262

Start 252, 257
StatusLine 158, 189
StopCountingAt 273
Stretch 166, 189, 252, 257, 292
Strikethrough 166, 189, 193, 252, 257, 292
Strings 189
StructAppAttrConfigFile 262
StructuredApplication 158, 189
StructuredApplicationForOpen 189
StyleOverride 170
StyleOverrides 280, 292
StyleOverrides(Read-Only) 189
SubFmtRule† 274
SubScriptShift 189
SubScriptSize 189
SubScriptStretch 189
SuffixRules(Read-Only) 272
SuperScriptShift 189
SuperScriptSize 189
SuperScriptStretch 189
Symbols 189
SymbolsList(Read-Only) 190
SystemVar(Read-Only) 294

T

TableTagging 272
Tabs 252, 257
TailArrow 203, 206, 208, 215, 217, 219, 221, 224, 226, 228, 230, 232, 235, 237, 239
TblAlignment 282, 284
TblBodyElement(Read-Only) 282
TblBodyFirstColor 282, 284
TblBodyFirstFill 282, 284
TblBodyFirstPeriod 282, 284
TblBodyNextColor 282, 284
TblBodyNextFill 282, 284

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

TblBodyNextPeriod 282, 285
TblBodyRowRuling 282, 285
TblBodyRowRulingPeriod 282, 285
TblBodyShadeBy 282, 285
TblBottomRuling 282, 285
TblCatalogEntry 285
TblCatalogEntry(Read-Only) 282
TblCellBottomMargin 282, 285
TblCellLeftMargin 282, 285
TblCellRightMargin 282, 285
TblCellTopMargin 282, 285
TblColRuling 282, 285
TblColRulingPeriod 282, 285
TblColWidths 282
TblElement(Read-Only) 282
TblFnCellPosition 190
TblFnCellPrefix 190
TblFnCellSuffix 190
TblFnCustNumString 163, 190
TblFnFmt 190
TblFnNumComputeMethod 163
TblFnNumStyle 164, 190
TblFnPosition 190
TblFnPrefix 190
TblFnSuffix 190
TblFooterElement(Read-Only) 282
TblHeaderElement(Read-Only) 283
TblHFColor 283, 285
TblHFFill 283, 285
TblHFRowRuling 283, 285
TblHFSeparatorRuling 283, 285
TblInitNumBodyRows 285
TblInitNumCols 285
TblInitNumFRows 285
TblInitNumHRows 285
TblLastBodyRuling 283, 285
TblLeftIndent 283, 285
TblLeftRuling 283, 285
TblNumbering 283, 285
TblNumCols(Read-Only) 283
TblNumRows(Read-Only) 283
TblOtherBodyRowRuling 283, 285
TblOtherColRuling 283, 285
TblPlacement 283, 286
TblRightIndent 283, 286
TblRightRuling 283, 286
TblSpaceAbove 283, 286
TblSpaceBelow 283, 286
TblTag 283, 286
TblTitleElement(Read-Only) 283
TblTitleGap 283, 286
TblTitlePosition 283, 286
TblTitleSelected(Read-Only) 284
TblTopRuling 284, 286
TblWidth(Read-Only) 284
Text 199, 235, 237, 270, 278
Text(Read-Only) 252
TextFmtRules(Read-Only) 272
TextLineType 211, 222, 224, 237
TextLoc 284
TextLoc(Read-Only) 200, 203, 240, 241
TextRange 170, 252
TextRange(Read-Only) 268, 287, 288, 289, 290, 293
TextSelection 190
Theta 206, 212
TiAutomaticUpdate 287, 288, 289, 290
TiByRows 290
TiClientData 287
TiClientName 287
TiClientSource 287
TiClientType 287
TiEOLisEOP 289
TiFile 287, 288, 289, 290

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

TiFileModDate 287, 288, 289, 290
TiFlowName 288
TiFlowPageSpace 288
TiFormat 288
TiHeadersEmpty 290
TiIsUnresolved 287, 288, 289, 290
TiLocked 288, 289, 290
TiMacEdition(Read-Only) 288, 289, 290
TiMainFlow 289
TintBaseColor 168
TintPercent 168, 203, 206, 208, 212, 215, 217, 219, 222, 224, 226, 228, 230, 232, 235, 237, 239
TiNumCols 290
TiNumHeaderRows 290
TiNumSeparators 290
TiRemoveOverrides 289
TiRemovePageBreaks 289
TiSeparator 290
TiTblTag 290
TiTextEncoding(Read-Only) 289, 290
TmpDir(Read-Only) 263
TomboMarks 187
TopMargin 190
TopRowSelection(Read-Only) 284
TopSeparator 252, 257
TrackChangesOn 190
Transient 241
TrapwiseCompatibility 190
TypeOfDisplayText 158

Untouchable 191
UseBkColor 169, 170
UseCapitalization 166
UseChangeBar 166
UseColor 167
UseFontAngle 167
UseFontFamily 167
UseFontSize 167
UseFontVariation 167
UseFontWeight 167
UseInitialStructure 158, 188, 191
UseInitialStructureOfAutoInsertedElements 158
UseKernX 167
UseKernY 167
UseNextTag 253, 257
UseOutline 167
UseOverline 167
UsePairKern 167
UsePosition 167
UserHomeDir(Read-Only) 263
UserLogin(Read-Only) 263
UserName(Read-Only) 263
UserSettingsDir 263
UserString 169, 269
UseSepOverride 170, 191, 280, 293
UseShadow 167
UseStretch 167
UseStrikethrough 167
UseUnderlining 167

U

Underlining 166, 191, 253, 257, 292
UndoFDKRecording 263
Unique(Read-Only) 164, 240, 258, 268, 288, 289, 290

V

ValidationFlags(Read-Only) 269
ValidHighestLevel 272
VarFmt 293
Variables 191

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

VersionMajor(Read-Only) 263
VersionMinor(Read-Only) 263
ViewBorders 191
ViewDisplayUnits 191
ViewFontSizeUnits 191
ViewFormattingBar 263
ViewGrid 191
ViewGridUnits 191
ViewNoGraphics 191
ViewOnlyDeadCodes 191
ViewOnlyMenuBar 190, 191
ViewOnlySelect 191
ViewOnlyWinBorders 158, 191
ViewOnlyWinMenuBar 191
ViewOnlyWinMenubar 158
ViewOnlyWinPalette 191
ViewOnlyWinPopup 191
ViewOnlyWinpopup 158
ViewOnlyXRef 192
ViewPageScrolling 192
ViewQuickAccessBar 263
ViewRulers 192
ViewRulerUnits 192
ViewTextSymbols 192
VolNumComputeMethod 164, 192
VolumeNumber 164, 192
VolumeNumStyle 164, 192
VolumeNumText 164, 192

W

WesternFamily 169
WesternFontPlatformName 165
WesternFontPostScriptName 165
WesternShift 169
WesternSize 169

WideRubiSpaceForJapanese 192
WideRubiSpaceForOther 192
Width 203, 206, 208, 213, 215, 217, 219,
222, 224, 226, 228, 230, 232, 235, 237, 239
Width(Read-Only) 276, 280, 293
WindowSystem(Read-Only) 263

X

XmlApplicationForBookComponent 164
XmlDoc 158, 193
XmlDocType 158, 193
XmlEncoding 158, 193
XmlFileEncoding 158, 193
XmlPublicId 159, 193
XmlStandAlone 159, 193
XMLStructuredApplicationList 263
XmlStyleSheet 159, 193
XmlStyleSheetList 159, 193
XmlSystemId 159, 193
XmlUseBOM 159, 194
XmlVersion 159, 194
XmlWellFormed 159, 194
XRefFile 170, 171
XRefFmt 170
XRefIsUnresolved 171
XRefSrcText 170, 171
XSLTProcessors 194, 263
XSLTTransformationScenarioFile 194

Y

YakumonoType 253, 257
Yellow 168

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Z

Zoom 194