

Adobe[®] FrameMaker 6.0



Using Filters



Contents

Introduction	About filters	3
	How filters work	3
	Supported formats	4
	Converting from unsupported formats	7
	Filter files	7
	Preparing files for conversion	7
	General importing issues	7
	Using graphics filters	8
	Document and text filter issues	11
	About Asian-language filters	13
	Converting files	16
	Converting FrameMaker documents	17
	Memory Requirements	17
Maker Interchange Format Filter Features	To MIF	18
	From MIF	23
Converting files using a UNIX window	Command line and filterbatch	30
	dcatomif filter	32
	iaftomif filter	34
	Converting IAF documents to MIF	39
	trofftomif filter	39
	fmimageroimage filter	41



Introduction

FrameMaker[®] product filters are programs that convert document and graphic files from one format to another. A document file contains information created with a word processor or publishing application; a graphic file contains a picture.

About filters

There are three types of filters:

- Document filters convert document files that were created with other applications into FrameMaker documents. The files may contain graphics. You convert document files by opening them with the FrameMaker product Open command or importing them with the Import command.

Through the Save As command, the document filters also export FrameMaker documents to other applications.

- Text filters convert text files between different character-encoding formats.
- Graphic filters import graphics created with other applications into FrameMaker documents. Through the Windows Clipboard, the graphic filters export FrameMaker graphics to other applications in BMP or WMF format. Through the Macintosh Clipboard, the graphic filters export graphics to other applications in PICT format. When you save a document to HTML, graphics can be exported to JPEG, GIF, or PNG.

FrameMaker and FrameMaker+SGML support all filters described in this manual. FrameViewer includes only import filters.

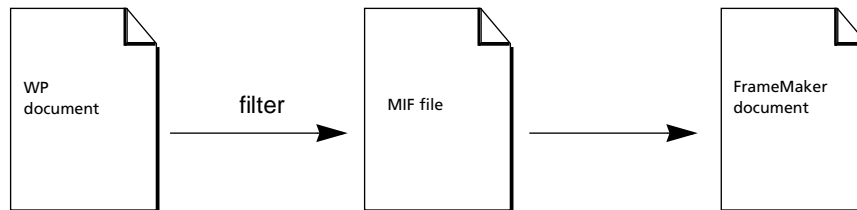
How filters work

Every software application has its own way of representing information (called an internal format). A filter converts one format to another. To facilitate filtering, some applications support interchange formats. For example, Rich Text Format (RTF) is an interchange format supported by many word processing applications. Instead of converting directly from one internal format to another, it's easier to convert from one format to the interchange format, and then from the interchange format to the other internal format. For example, FrameMaker products include a filter to translate to RTF format. You can open the resulting RTF file with any word processor that supports the format.

Maker Interchange Format (MIF)

FrameMaker products provide an interchange format called Maker Interchange Format (MIF). For information about MIF, see the online manual *MIF Reference*. Many of the FrameMaker document filters convert word processing formats to an intermediate MIF file. FrameMaker products include an internal filter that converts MIF to the FrameMaker document format.

Below is an illustration of the process for converting a word processing document to a FrameMaker document.



Graphic formats

FrameMaker filters convert graphics to FrameVector format (the internal format for representing graphic objects in FrameMaker products) or to image (bitmap) format.

All graphics in a file converted to FrameVector or image format convert as a single graphic object. You can reposition and resize the object, but you cannot change individual graphic objects within it. There is one exception, some PICT images can be ungrouped (see the *FrameMaker User Guide* for additional information).

Supported formats

You can convert documents and graphics in the formats listed in the following tables. The platforms are Windows (W), Macintosh (M), and UNIX (U). The .x in the version column indicates that all minor releases are supported.

Document import filters

Format	Platform	Versions supported
Text	W,M,U	ASCII, Windows ANSI, Macintosh Text
DCA/RTF, DisplayWrite	W,M,U	
Interleaf	W,U	4, 5, 6 IAF 6.4, 8.0, 9.6
Lotus 123	W,M	4.0, 5.0
Lotus Ami/Word Pro	W,M,U	1, 2, 3.0, 3.1
MacWrite	M	4.x, 5.x
MacWrite II, Pro	M	
Microsoft Word	W,M,U	Windows 1.x, 2.x, 6.x, 7.x, Word97, Word2000 Macintosh 3.x, 4.x, 5.x, 6.x, Word 98 DOS 4.0, 5.x
Microsoft Excel	W,M,U	5.0/95
MIF	W,M,U	1.x, 2.x, 3.x, 4.x, 5.x
RTF	W,M,U	1.2, 1.3
troff	U	
Ventura (Corel)	W	3.0, 4.0
WordPerfect	W,M,U	Windows 5.x, 6.x Macintosh 1.x, 2.x, 3.x

Document export filters

Format	Platform	Versions supported
ASCII Text	W,M,U	ASCII, Windows ANSI, Macintosh Text
Hypertext Markup Language (HTML)	W,M,U	
Extensible Markup Language (XML)	W,M,U	
Interleaf	U	5.x (IAF 6.4)
MacWrite	M	4.x, 5.0
MacWrite II, Pro	M	
Microsoft Word	W,M,U	Windows 6.x, 7.x Macintosh 4.x, 5.x, 6.x
Portable Document Format (PDF)	W,M,U	3, 4
MIF	W,M,U	4.x, 5.x
RTF	W,M,U	1.2, 1.3
WordPerfect	W,M,U	Windows 5.x Macintosh 3.x

Graphic import filters

Format	Platform	Versions supported
AutoCAD Drawing Interchange Format (DXF, DWG)	W,M,U	10, 11, 12, 13
BMP	W	Windows 3.x, Windows 95
CompuServe Graphics Interchange Format (GIF)	W,M,U	87a, 89a Transparent backgrounds are not supported.
Computer Graphics Metafile (CGM)	W,M,U	ANSI/ISO 8632:1992
CorelDraw (CDR)	W,M,U	3, 4, 5, 6, 7
Digital Research Graphics Environment Manager (GEM)	W,M,U	3
EPS, EPSF, EPSI	W,M,U	
Hewlett-Packard Graphics Language (HPGL)	W,M,U	2
Initial Graphics Exchange Specification (IGES)	W,M,U	CALS Classes I-IV, Release 5.1
JPEG	W,M,U	
Macintosh Quickdraw (PICT)	W,M,U	1, 2
MacPaint	W,M,U	PackBits compression
Micrografx Drawing File Format (DRW, GRF)	W,M,U	Draw 3.0, Designer 3.1, Charisma 2.1
PCX/DCX	W,M,U	1-bit, 4-bit, 8-bit, 24-bit
Portable Network Graphics (PNG)	W,M,U	
Sun Raster (rf)	W,M,U	
TIFF	W,M,U	5, 6 (CMYK separations)
Windows Metafile (WMF)	W,M,U	Windows 3.x, Windows 95
WordPerfect (WPG)	W,M,U	1, 2
X Windowing System bitmap (XWD)	W,M,U	Release X11
XBM	U	
CCITT G4	W,M,U	Type 1
RGB-SGI	U	
Enhanced Metafile (EMF)	W	
QuickTime Movie	M	

Converting from unsupported formats

If you need to convert files that are in formats for which FrameMaker products have no specific filter, you may be able to save the files to a format recognized by one of the filters (for example, RTF). You can then convert them from that format into FrameMaker documents. Check whether your application can save files in a different format. This might be an option in the application's Save As command, or you might have to use a separate program to convert the file. An application's conversion programs are often on a separate disk in the installation package. To find out where the conversion programs are, refer to your application's reference manual.

Filter files

During installation, you can choose whether to install filter files. The filters directory is located in the directory where FrameMaker is installed. If you choose to install the filters at a later date, you'll need to run the Installer again. For more information about installing FrameMaker products (including filters), see the installation manual.

Preparing files for conversion

The filters convert the specified features of the source files into equivalent FrameMaker product features. For best results, follow these suggestions for preparing files before converting them:

- Avoid using tricks that force the application to simulate functions it does not naturally perform. For example, using negative indents to create side headings can make conversion more difficult.
- Use defined styles rather than local format overrides whenever possible.
- The filters assign the first column layout in the source file to all body pages in the converted document. It is easier to touch up converted documents that have more than one column layout if the most frequently used column layout is the first one in the source file.
- Create a template in FrameMaker that has predefined paragraph and character styles, tables definitions, and page layout for when you convert a file to FrameMaker. If the FrameMaker template is set up correctly, the majority of issues regarding autonumbering, bullets, indenting, and page layout should be resolved. After you import the file, use the File > Import > Formats command to apply the FrameMaker template to the imported file. If your Word file and FrameMaker template use different styles, delete all paragraph and character styles, apply the FrameMaker template and then apply the FrameMaker styles to your text after you import the file into FrameMaker. For information on using templates and importing formats, see Chapter 12, "Templates," in the *FrameMaker User Guide*.

General importing issues

When importing documents into FrameMaker, note the following:

- Customized bullet symbols used in bulleted lists are sometimes converted to the default bullet symbol (•) when a document is imported or exported. If this occurs and you would like to change the formatting of your bullets, see "Formatting text as bullets" in Chapter 4, "Text Formatting," of the *FrameMaker User Guide*.

- If a document includes pages in both portrait and landscape page layout, the page layout orientation that appears first in the original document will be used to format the entire converted document.
- Before exporting a FrameMaker document on the Macintosh, remove any EPS graphics from the document.
- If you import document files (such as Microsoft Word or WordPerfect) or RTF files that include tables, you may have difficulty formatting the resulting tables in the FrameMaker document. Initially, you may have trouble applying format changes via the table designer. To use the table designer for a given table, first select the entire table. Then choose Custom Ruling & Shading from the Table menu. In the dialog box that appears, apply From Table for Custom Cell Ruling and Custom Cell Shading. Now you can apply formatting via the table designer.

Using graphics filters

FrameMaker uses IsoDraw filters to import and export graphics files. However, you can use different graphics filters if you desire. Inso filters are also provided with version 6 of FrameMaker products. If you want to use the Inso filters, you must install them and uninstall the corresponding IsoDraw filters by following the platform-specific instructions. See “Changing default CGM filters” on page 10.

Filters for CGM files

In version 6, the installed filters for CGM graphic files are IsoDraw Converter filters. The IsoDraw filters included with FrameMaker products support a wide variety of CGM formats, including CGM 1, 2, 3, and 4 for import; CGM 1, 2, and 3 for export; and many CGM profiles customized for use by specific industries.

Important: *IsoDraw also supplies HPGL, IGES, and DXF filters. However, these filters have not been fully tested with FrameMaker products and may have unpredictable results.*

- Most fonts are now mapped correctly when CGM files are imported. Font mapping can be customized via the optcgm.prf file. For more information, see “Modifying preferences for IsoDraw filters” on page 9.
- Rounded line caps and line joins of thick lines in imported graphics are rendered incorrectly when importing a CGM file; the junctions appear as square. FrameMaker supports only square line joinings.
- When importing CGM graphics, the IsoDraw filters use background colors by default. It is often useful to consider the background as part of the graphic. For example, a graphic might use a white background to attain a specific amount of white space around the graphic.
- You can change the preferences for CGM import to ignore the background color. This might be necessary if you have a black background and black lines or text in the graphic. Note that when you ignore the background, FrameMaker fits the graphic’s anchored frame around the art, not around the area usually taken by the background.
- To change the preference, open OptCGM.prf (Macintosh and UNIX) or optcgm.prf (Windows) and find the entry CGM_Background. Change it from CGM_Background USE to CGM_Background IGNORE, then save the file as text. For more information about changing CGM filter preferences, see “Modifying preferences for IsoDraw filters” on page 9.

- If you plan to export and then import the same CGM file (referred to as “roundtripping” the file), Adobe recommends that you use filters by the same manufacturer for both operations. For example, if you use an IsoDraw filter to export a file, you should use an IsoDraw filter to import the file back into a FrameMaker product.

Filters for TIFF files with alpha channel information

FrameMaker 6.0 supports only 1 byte TIFF files with alpha channel information. The TIFF filter strips the alpha channel information and displays the image. If the TIFF has extrasamples set to 2, then the filter multiplies the extrasamples and image pixels. This may or may not be correct for some TIFFs.

Modifying preferences for IsoDraw filters

To modify preferences for an IsoDraw filter, use a text editor to open a preference file for that filter. Follow the instructions in the preference file for modifying preference settings, and then save the file as text and close. There are no preference files for HPGL filters.

The preference files and their locations are different for the different platforms, as follows:

- (Windows) IsoDraw preference files are located in the filters folder in the main FrameMaker product folder. The preference files are `optcgm.prf` for the CGM filter, `optdxf.prf` for the DXF filter, and `optigs.prf` for the IGES filter.
- (Macintosh) IsoDraw preference files are located in the IsoConverter folder, located in System Folder:Preferences. The preference files are `OptCGM.prf` for the CGM filter, `OptDXF.prf` for the DXF filter, and `OptIGS.prf` for the IGES filter.
- (UNIX) If you installed IsoDraw filters on a network and you want to modify preferences for all users, modify the preference files in the network installation of `$FMHOME/fmunit/IsoConverter`. If you want to modify preferences for a single user, copy the IsoConverter folder to the user’s installation of `$HOME/fmunit/`, then modify the preference files there. If you cannot locate the IsoConverter folder in the network or user installation, modify the preference files in `/tmp/IsoConverter`.

The preference files are `OptCGM.prf` for the CGM filter, `OptDXF.prf` for the DXF filter, and `OptIGS.prf` for the IGES filter.

- Inso filters for CGM import and export are also provided with version 6 of FrameMaker products. If you want to use the Inso filters, you must install them and uninstall the corresponding IsoDraw filters by following the platform-specific instructions in “Changing default CGM filters” on page 10.

To generate error log messages for Isodraw CGM filters:

- 1 Open the `OptCgm.prf` file in a text editor and search for the following line:

```
Logfile : 0
```

- 2 Edit this line as follows:

```
Logfile : 1
```

- 3 Save the `OptCgm.prf` file. If the Isodraw CGM filter finds any errors while importing from CGM, the errors will appear in the file `xxxxx.LOG`, where `xxxxx` is the CGM filename.

Changing default CGM filters

See “Filters for CGM files” on page 8 for information on IsoDraw filters, the default CGM filters for version 6. To use the Inso filters that are also included, you must install them and uninstall the IsoDraw filters. See the information that follows for your platform.

(Windows) To enable Inso filters and disable the corresponding IsoDraw filters:

1 In a text editor such as Notepad, open the maker.ini file in the FrameMaker 6 folder (for FrameMaker) or the fmsgml.ini file in the FrameMaker+SGML 6 folder (for FrameMaker+SGML).

2 In the Filters section of the file, locate the lines referring to the Inso filters you want to enable.

Each line contains the name of the filter (CGM) and the term GFXImport (for an import filter) or GFXExport (for an export filter).

3 Remove the semicolon (;) at the beginning of each line to enable the filters.

4 In the APIClients section of the file, locate the lines referring to the IsoDraw filters that correspond to the Inso filters you just enabled.

Each line contains the name of the filter (CGM Import or CGM Export) and the term IsoConverter.dll. For example, if you have installed the Inso CGM import and export filters (CGM IMAG GFXImport and CGM IMAG GFXExport), locate the IsoDraw CGM Import and CGM Export filters.

5 Insert a semicolon at the beginning of each line to disable the filter.

6 Save and close the maker.ini or fmsgml.ini file, and restart the FrameMaker product.

(Macintosh) To enable Inso filters and disable the corresponding IsoDraw filters:

1 In your FrameMaker product installation folder, locate the System Extras:Filters folder.

2 Drag all the files in that Filters folder into your FrameMaker product’s Filters:INSO folder. (The names of the files begin with CGM.)

3 Create a new folder named IsoDraw Filters Disabled in the System Extras folder.

4 Locate the IsoDraw Converter file in your FrameMaker product’s Filters folder, and drag it to the System Extras:IsoDraw Filters Disabled folder.

5 Restart the FrameMaker product.

(UNIX) To enable Inso filters and disable the corresponding IsoDraw filters:

1 Locate the \$FMHOME\fminit\filterlist file and do one of the following:

- If you are enabling Inso filters for all users on a network, open the file in the network installation’s fminit directory.
- If you are enabling Inso filters for a single user, copy the filterlist file to the user’s fminit directory and open the copy.

2 In the FilterList section of the filterlist file, locate the lines referring to the Inso filters you want to enable.

Each line contains the name of the filter (CGM) and the term GFXImport (for import filters) or GFXExport (for export filters).

3 Replace the parentheses enclosing these lines with angle brackets (< >) to enable the filters.

4 In the Filterlist Specification section of the filterlist file, locate the lines referring to the IsoDraw filters that correspond to the Inso filters you just enabled.

Each line contains the name of the filter (CGM) and the term GFXImport (for import filters) or GFXExport (for export filters). For example, if you have enabled the Inso CGM import and export filters (CGM IMAG GFXImport and CGM IMAG GFXExport), locate the IsoDraw CGM import and export filters (CGM ISO GFXImport and CGM ISO GFXExport).

5 Replace the angle brackets enclosing these lines with parentheses to disable the IsoDraw filters, and then save and close the filterlist file.

6 Restart the FrameMaker product.

Document and text filter issues

FrameMaker provides document filters for importing (and in some cases exporting) Microsoft Word, WordPerfect, Ami Pro, RTF and Interleaf documents. FrameMaker also provides filters for importing and exporting text files.

Microsoft Word filters

You can now import Microsoft Word 97/98/2000 documents. To use the new Word filters, choose Microsoft Word 97/98/2000 when you see the list of file types upon opening. When importing Word documents, note the following limitations:

- The cell shading in Word tables are mapped to the nearest value allowed by FrameMaker table cell shading values (3, 10, 30, 70, 90, white, or None).
- Macros within Word documents are not converted.
- Equations within Word documents convert as WMF insets.
- Bookmarks within Word documents become cross-reference markers.
- Annotations in Word documents become conditional text with the condition “Comment” when imported by reference.
- Hidden text in Word documents becomes non-hidden conditional text with the condition “Hidden” when imported.
- If a Word document includes a table of contents with toggled field codes stored in the Word document, the table of contents in the converted document may contain incorrect page numbers.
- Some Word documents may contain extra tabs after conversion. You can search for tabs by choosing Edit > Find/Change and typing /t in the Find box.

- The Microsoft Word 97/98/2000 import filter does not support CGM graphics, OLE graphics, linked objects, Word art, Word drawing objects, captions, and autoshapes. However, Word art, Word drawing objects, and autoshapes will be converted if they are inserted using the Insert > Object > Microsoft Word Picture command in Word. Similarly, captions will be converted if they are inserted using the Insert > Captions command in Word. On the Macintosh, this filter does not import JPEG and EMF graphics from Word documents.
- The Microsoft Word 97/98/2000 import filter does not import multilevel autonumbering correctly. If the Word document was saved using Fast Save, the resulting numbering will not increment in the FrameMaker document. If you did not save with Fast Save, the numbering increments, but restarts (returns to 1, for example) at inappropriate places. To fix the autonumbering, open the paragraph designer and edit the autonumbering string for the paragraph formats that have autonumbering.

***Note:** In the initial release of Word 97, the Word 6/95 option in the Save As menu saves a document in Rich Text Format (RTF). To open a document in a FrameMaker product, you must use the Microsoft RTF filter. If you choose the Microsoft Word 97/98/2000 filter instead, an alert message will appear stating that the document will not be converted. Microsoft has released an update to the filter, which creates a Word 6 binary file rather than an RTF file. You can download this filter at www.microsoft.com/word/enhancements/wrd6ex32.asp.*

- All FrameMaker graphics which are placed in an anchored frame are exported to Word 6/RTF on all platforms. Graphics which are not in anchored frames will not be exported.
- For export to Word and RTF, the formatting of your document converts properly. However, the named styles may not export correctly; the style names appear in the user interface, but when you apply such a named style the selected text appears the same as “normal” text.

Exporting frame native graphics to Word6/RTF

Note the following limitations when a FrameMaker file containing native graphic objects such as rectangles and circles is exported to Microsoft Word6.0/95 or Microsoft RTF:

- 3 point line widths may appear as 2.5.
- The dashed line styles may appear different after export.
- Arrow lines are not converted as a single object. When edited in the Microsoft Word Picture, arrow lines will appear as separate line and arrow heads.
- Some arrow styles in Word may not correspond to what is seen in FrameMaker.
- Some curves, ovals, ellipses, and freehand drawings may appear jagged.
- Dashed curves, ovals, ellipses, freehand drawings, polylines, and polygons are converted into solid style.
- Thick lines are converted to rectangles.
- Pens and tints are not supported.

- Graphic text kerning and stretching are not supported.
- Some native graphics exported from FrameMaker are not suitable for further editing in Microsoft Word Picture.

Microsoft Excel filters

To import a document created in Office 97 Excel into a FrameMaker product, you must save the document as Microsoft Excel 5.0/95 Workbook. If you try to import a document saved in the default format (Microsoft Excel Workbook), the document will not convert and you will see the message: “The filter encountered an error and could not complete the translation.”

WordPerfect filters

In WordPerfect 6 and 7 documents, import of autonumbering through paragraph outline is supported, but import of autonumbering through counters is not. If a WordPerfect document contains both kinds of autonumbering, the autonumbering may be incorrect after conversion. Right indents in WordPerfect documents may not import correctly.

If you are using WordPerfect 8 or a later version, save the file in WordPerfect 6 or 7 format before importing it into FrameMaker.

About Asian-language filters

The following formats are supported for importing Japanese text:

- Text format
- Japanese Microsoft Word 6.0, Office 95/Word 7.0, and Word 97/Word 8 formats
- Ichitaro 5.0 and 6.0 Text Import Filter (Windows, Macintosh, and Solaris UNIX only)
- RTF
- Word 97/98/2000

The following formats are supported for exporting Japanese text:

- Text format
- PDF format (Windows and Macintosh only)
- HTML format
- XML format
- RTF (Macintosh, Windows, and Solaris UNIX)

The following text encoding methods are supported for importing or exporting Japanese text:

- JIS (a common encoding method for Japanese e-mail)
- Shift-JIS (the most common encoding method on Japanese Windows and Macintosh systems)
- EUC (the most common encoding method on Japanese UNIX systems)

Japanese filters support the JIS X 0208-1990 character set. The Ichitaro and RTF Japanese filters do not support certain symbols and accented Roman characters outside of this character set. The Word 97 filter will import certain symbols as well as Russian and Greek characters as unrecognizable characters. Applying a Japanese font to these unrecognizable characters will usually correct this.

All filters in Japanese versions of FrameMaker products support conversion of English documents.

All graphics filters are supported in Japanese versions of FrameMaker products, but conversion of Japanese text contained within graphics is not supported.

If you import files other than text files into Japanese versions of FrameMaker products on a different platform from the one on which the files were created, the imported files may display in incorrect fonts. To resolve this problem, change the font in the Paragraph or Character Designer dialog boxes. Japanese autonumbering is supported by FrameMaker products but not by Japanese text filters.

If you import a document that uses a Japanese numbering system, such as iroha or gojuuon ordering, these numbering systems can be reapplied using Numbering properties in the paragraph designer.

Ichitaro filters

To import an Ichitaro 5.0 or 6.0 Windows file, follow the standard import procedures as described in the *FrameMaker User Guide*. You do not need filename extensions for the FrameMaker product to recognize these files. However, if you want to use filename extensions, for Ichitaro 5.0 files use the extension .jaw, and for Ichitaro 6.0 files use the extension .jbw.

To import files created with Ichitaro 7 or 8, first save the files as Ichitaro 5.0 (.jaw) or 6.0 (.jbw) files.

When importing Ichitaro documents, tables created using commands from the Keisen (Ruling) menu will convert into a FrameMaker table. Also, when the outer frame is set to mendori (rounded corners), the enclosed text imports as a table. However, if a table breaks across pages, the text does not import as a table. Also, Ichitaro 8 tables created using Keisen > Create Table may not import correctly.

Side notes, contents, indexes, references, locks, calculations, and renban (autonumbering) import as normal text. Paragraph hierarchy data is lost, although the resulting paragraph indents are retained.

Accented Roman characters lose their accents after importing from or exporting to Ichitaro documents.

Microsoft Word filters in Japanese versions

If you use the Microsoft Word filter to import a file containing rubi (furigana) characters into Japanese versions of FrameMaker products, the rubi will appear in the imported file before the characters that they support (oyamoji). In some instances, spurious characters may appear.

If you use the Microsoft Word filter to import a Word document containing style formatting, the style formatting imports as paragraph formatting in Japanese versions of FrameMaker products. The formatting data are preserved, but the format tag name might be corrupted. If this occurs, choose New Format from the Commands menu in the paragraph designer, then rename and save the format.

You cannot export a document with Japanese text to Microsoft Word. Instead, export the document to RTE, and then open the RTE file in Microsoft Word.

If you use the Microsoft Word filter to import a Word 97/98/2000 document, the paragraph formats will specify a Roman font as the default font. Also, character formats that use a Symbol font will be changed to use a Western Symbol font, rather than a Japanese one. Do the following:

- 1 Create a combined font, if desired.
- 2 In the Paragraph Designer (for paragraph formats) or Character Designer (for character formats), change Default Font Name to the font name you want.
- 3 Change Language to Nihongo.
- 4 If needed, correct the autonumbering format for the paragraph format.
- 5 Click Update All to apply these changes to all occurrences of the format.

RTF filters in Japanese versions

FrameMaker products for the Windows, Macintosh, and Solaris (UNIX) platforms offer two filters for importing RTF documents: Microsoft RTF and RTF Japanese. You will usually see the best results by using the default filter for your system—RTF Japanese if you have a Japanese operating system, otherwise Microsoft RTF. However, you should keep the following in mind:

- When importing and exporting the same file, you should use the same filter in both directions.
- When filtering both Japanese and Western documents, if you want the same formatting for both types, you should use the same filter for both types.
- For Japanese documents, the Microsoft RTF filter is for import only.
- The different filters support different sets of features. The Microsoft RTF filter can import embedded graphics, and the RTF Japanese filter supports table shading and ruling and also multiple-column documents.
- For imported RTF documents, rubi (furigana) appear within square brackets in the FrameMaker document. When a FrameMaker document is saved as RTF, the rubi are exported as normal text.
- When saving as RTF or when opening an RTF file in FrameMaker on the Macintosh, if the document is very large, your system may develop a memory error. Try dividing the document into smaller files before creating the RTF file, or set the FrameMaker application memory to the recommended 24 MB.

Filters for Chinese and Korean text

FrameMaker products cannot import or export documents with Chinese or Korean text to RTF.

FrameMaker can import and export text files encoded in Big5, EUC-CNS (also known as EUC-TW) for Traditional Chinese, GB (also known as EUC-CN) and HZ (HZ-GB-2312) for Simplified Chinese, and EUC-KR (also known as Wansung) for Korean.

Text filters that come with FrameMaker and FrameMaker+SGML support English text files only. Chinese and Korean text filters are not supported. For more information see the *Chinese and Korean Features* online manual.

Converting files

You can convert document and text files by either opening or importing them. Convert by opening when you want to convert an entire file. Convert by importing when you want to copy a file into an existing FrameMaker product document.

Converting by opening

To convert a file by opening it:

- 1 Start the FrameMaker product.
- 2 Choose File > Open.
- 3 Select the file you want to open and click Open.
- 4 If the Unknown File Type dialog box appears, select the appropriate application or file format and click Convert.

It may take a few moments to convert a large file. When the conversion is complete, the FrameMaker document opens.

Converting by importing

The Import File dialog box contains the settings Import by Reference and Copy into Document. You can import graphic and text files either by reference or copying into the document.

To convert a file by importing it:

- 1 Start the FrameMaker product and open a document.
- 2 Place the insertion point where you want the imported document or graphic to appear.
- 3 Choose File>Import>File from the File menu.
- 4 In the scroll list, select the name of the file you want to import.
- 5 Select Import by Reference or Copy into Document, and click Import.
- 6 If the Unknown File Type dialog box appears, select the appropriate application or file format and click Convert.

The file is converted and imported into the document. (For more information on import options, see the *FrameMaker User Guide*.)

Document files appear as text insets if imported by reference. Graphic files are converted and placed into an anchored frame at the location of the insertion point in the FrameMaker document. If you're importing a bitmap, a dialog box appears so you can specify the bitmap's scale.

Converting FrameMaker documents

To export a FrameMaker document to another application:

- 1 Choose Save As from the File menu.
- 2 Type a new name for the file in the Save Document As dialog box.
- 3 Choose the format you want from the Format pop-up menu.
- 4 Click Save.

Important: When you create FrameMaker documents that you intend to convert to another format, paragraph tags must not include the bracket symbols [] or < >. These symbols disable cross-references and variables.

Memory Requirements

Filters run within the FrameMaker product's application memory allocation. You cannot filter a document larger than this setting. If a document is too large for the filter, close other files to free memory or open the document in its original application and split it into smaller pieces. Also, make sure you have extra free disk space for importing large files (this space is needed to create temporary files).

Maker Interchange Format (MIF) Filter Features

To MIF

The following table shows what is converted from other applications when using MIF to import into a to FrameMaker product:.

Character Attributes	YES	NO	N/A	Comment
Bold / Weight	X			
Code Pages	X			
Color:	X			
Background Text			X	
Foreground Text	X			
Comment	X			conditional text
Drop Cap			X	
Fonts:	X			
Point Size	X			
Proportional / Serif			X	
Typeface	X			
Hidden Text	X			conditional text
Initial Caps			X	
Italic / Slant	X			
Kerning	X			
Languages	X			
Outline		X		
Redline:	X			conditional text
Deleted	X			conditional text
Inserted	X			conditional text
Shading:			X	
Color			X	
Pattern			X	
Shadow	X			

Character Attributes (con't)	YES	NO	N/A	Comment
Small Caps	X			
Soft Hyphen	X			
Strike Out	X			
Subscript	X			
Superscript	X			
Upper Case/All Caps	X			
Underline:	X			
Continuous Single	X			
Continuous Double	X			
Word only Single		X		
Word only Double		X		
Dotted Underline			X	

Line Attributes	YES	NO	N/A	Comment
Line Height	X			
Line Numbering			X	

Paragraph Attributes	YES	NO	N/A	Comment
Alignment:	X			
Center	X			
Full Left	X			
Full all Lines			X	
Left	X			
Partial			X	
Right	X			
Borders:			X	
Color			X	
Indent:	X			
Hanging	X			
Left	X			
Right	X			
Keep Lines Together	X			

Paragraph Attributes (con't)	YES	NO	N/A	Comment
Keep With Next	X			
Line Spacing	X			
Paragraph Numbering	X			
Shading:			X	
Color			X	
Pattern			X	
Space Before/After		X		
Tabs:	X			
Center	X			
Decimal	X			
Leaders	X			
Left	X			
Relative vs. Absolute			X	
Right	X			
Bar			X	
Widows/Orphans		X		
Relative vs. Absolute			X	
Right	X			
Bar			X	
Widows/Orphans			X	
Borders:			X	
Color			X	
Hard Page Breaks	X			
Headers and Footers:	X			added to master pages
All pages	X			
First Page Only	X			
Odd/Even Pages	X			
Margins	X			only affects master pages if encountered before text
Left	X			
Right	X			

Page Attributes	YES	NO	N/A	Comment
Top	X			
Bottom	X			
Header			X	
Footer			X	
Header Top			X	
Footer Bottom			X	
Orientation	X			
Landscape	X			see margins
Portrait	X			see margins
Page Numbering	X			
Paper Size	X			see margins
Shading			X	
Color			X	
Pattern			X	
Soft Page Breaks	X			

Document Attributes	YES	NO	N/A	Comment
Document Summary Info			X	
Password			X	
Protected			X	
Shading			X	
Color			X	
Pattern			X	

Columns and Tables	YES	NO	N/A	Comment
Columns:		X		
Balanced		X		
Borders:			X	
Color			X	
Column Spacing		X		
Hard Column Breaks			X	
Lines Between			X	

Columns and Tables (con't)	YES	NO	N/A	Comment
Newspaper (snaking)		X		
Parallel			X	
Shading:			X	
Color			X	
Pattern			X	
Soft Column Breaks			X	
Tables:	X			
Cell Alignment	X			
Horizontal	X			
Vertical			X	
Cell Borders:		X		
Color		X		
Cell Shading:		X		
Color		X		
Pattern		X		
Heading		X		
Merged Cells:	X			
Horizontal	X			
Vertical	X			
Row Height		X		
Table Alignment		X		
Table Border:		X		
Color		X		
Table Shading:		X		
Color		X		
Pattern		X		

Miscellaneous Attributes	YES	NO	N/A	Comment
Annotations			X	
Bookmarks			X	
Cross References	X			
Date/Time	X			

Miscellaneous Attributes (con't)	YES	NO	N/A	Comment
Endnotes			X	as footnotes
File Include		X		
Footnotes	X			
Formulas		X		
Frames/APO's	X			
Graphics:				
Caption			X	
Embedded Graphics		X		
Referenced Graphics		X		
Watermark			X	
Hypertext	X			
Indexes	X			
Macros			X	
Multi-Byte Characters	X			Shift-JIS only
OLE	X			
Outline			X	
Style Sheets	X			
Table of Authority			X	
Table of Contents	X			
Text Direction:			X	
Left To Right			X	
Right To Left			X	

From MIF

The following table shows what is converted from a FrameMaker product when using MIF to export into other applications:.

Character Attributes	YES	NO	N/A	Comment
Bold / Weight	X			
Code Pages	X			
Color:	X			
Background Text			X	
Foreground Text	X			

Character Attributes (con't)	YES	NO	N/A	Comment
Comment	X			conditional text
Drop Cap			X	anchored frames
Fonts:	X			
Point Size	X			
Proportional / Serif			X	
Typeface	X			
Hidden Text			X	
Initial Caps			X	
Italic / Slant	X			
Kerning		X		
Languages	X			
Outline			X	
Redline:	X			treated as conditional
Deleted	X			
Inserted	X			
Shading:			X	
Color			X	
Pattern			X	
Shadow	X			
Small Caps	X			
Soft Hyphen	X			
Strike Out	X			
Subscript	X			
Superscript	X			
Upper Case/All Caps	X			
Underline:	X			
Continuous Single	X			
Continuous Double	X			
Word only Single			X	
Word only Double			X	
Dotted Underline			X	

Line Attributes	YES	NO	N/A	Comment
Line Height	X			
Line Numbering		X		

Paragraph Attributes	YES	NO	N/A	Comment
Alignment:	X			
Center	X			
Full Left	X			
Full all Lines			X	
Left	X			
Partial			X	
Right	X			
Borders:			X	
Color			X	
Indent:	X			
Hanging	X			
Left	X			
Right	X			
Keep Lines Together			X	
Keep With Next	X			
Line Spacing	X			
Paragraph Numbering	X			
Shading:			X	
Color			X	
Pattern			X	
Space Before/After	X			
Tabs:	X			
Center	X			
Decimal	X			
Leaders	X			
Left	X			
Relative vs. Absolute			X	

Paragraph Attributes (con't)	YES	NO	N/A	Comment
Right	X			
Bar			X	
Widows/Orphans		X		

Page Attributes	YES	NO	N/A	Comment
Borders:			X	
Color			X	
Hard Page Breaks	X			
Headers and Footers:			X	
All pages			X	
First Page Only			X	
Odd/Even Pages			X	
Margins	X			
Left	X			
Right	X			
Top		X		
Bottom		X		
Header			X	
Footer			X	
Header Top			X	
Footer Bottom			X	
Orientation	X			
Landscape	X			
Portrait	X			
Page Numbering	X			
Paper Size	X			
Shading			X	
Color			X	
Pattern			X	
Soft Page Breaks			X	

Document Attributes	YES	NO	N/A	Comment
Document Summary Info			X	
Password			X	
Protected			X	
Shading:			X	
Color			X	
Pattern			X	

Columns and Tables	YES	NO	N/A	Comment
Columns:		X		
Balanced		X		
Borders:			X	
Color			X	
Column Spacing		X		
Hard Column Breaks			X	
Lines Between			X	
Newspaper (snaking)		X		
Parallel			X	
Shading:			X	
Color			X	
Pattern			X	
Soft Column Breaks			X	
Tables:	X			
Cell Alignment		X		
Horizontal		X		
Vertical		X		
Cell Borders:		X		
Color		X		
Cell Shading:		X		
Color		X		
Pattern		X		
Heading			X	
Merged Cells:	X			

Columns and Tables (con't)	YES	NO	N/A	Comment
Horizontal	X			
Vertical	X			
Row Height		X		auto height
Table Alignment		X		
Table Border:		X		
Color		X		
Table Shading:		X		
Color		X		
Pattern		X		

Miscellaneous Attributes	YES	NO	N/A	Comment
Annotations			X	
Bookmarks			X	
Cross References	X			
Date/Time	X			
Endnotes			X	
File Include	X			
Footnotes	X			
Formulas		X		
Frames/APO's	X			
Graphics:				
Caption		X	X	
Embedded Graphics		X		
Referenced Graphics		X		
Hypertext	X			
Indexes	X			
Macros			X	
Multi-Byte Characters	X			
OLE	X			
Outline			X	
Style Sheets	X			

Miscellaneous Attributes (con't)	YES	NO	N/A	Comment
Table of Authority			X	
Table of Contents		X		
Text Direction:			X	
Left To Right			X	
Right To Left			X	

Converting files using a UNIX window

Command line and filterbatch

The easiest way to filter a file is to open or import it with a FrameMaker product. On UNIX systems, most filters are separate UNIX programs that a FrameMaker product runs when you open or import a file with the corresponding format. You can also run these filters from a UNIX window, or use the `filterbatch` program to filter several files at once.

Other UNIX filters are applications created with the Frame Application Program Interface™ (API), which is part of the Frame Developer's Kit™. These filters are integrated into FrameMaker products, and run automatically when you import or open such files. You cannot run these filters from a UNIX window.

The UNIX filters directory

UNIX filter files are located in the product installation directory and are shared by all installed FrameMaker products. Some UNIX filters require setup files that specify details on how files are converted. For example, the `iaftomif` filter uses a setup file to map Interleaf fonts to FrameMaker product fonts. Filter setup files are located in `$FMHOME/fmunit/filters`. To customize filter setup files for your own use, you can create personal `filters` directories in other directories.

Important: FrameMaker searches for the `fmunit` directory in the following directories in this order: the directory in which you started the product, your home directory, and then `$FMHOME`.

The following table lists filter setup files and the filters that use them. Do not remove or edit any setup files unless this manual instructs you to do so.

File	Filter	Can be customized?
dca.am	dcatomif	Yes
lsi.am	dcatomif	Yes
iaf.am	iaftomif	Yes
miftoiaf.fonts	miftoiaf	Yes
patterns	miftoiaf	Yes
pictfonts.cfg	picttomif	No
txttomif.m4	is1tomif, rm8tomif	Yes
wmchar.cnv	wptomif	Yes
wmcolor.tab	wptomif	No
wmfont.cnv	wptomif	Yes
wmfont.dat	wptomif	No

Adding filename suffixes

When you open or import a file, FrameMaker looks for information in the file to determine the file type. If the information isn't found in the file, the filename suffix is used to determine the file type.

To add or change the suffix of a group of filenames, use the `add_extension` script. In a UNIX window, type:

```
add_extension suffix filelist
```

where *suffix* is the suffix for your file type and *filelist* is a list of filenames, separated by spaces, to which you want to add the suffix.

For example, to add the `.iaf` suffix to a group of Interleaf IAF files, type:

```
add_extension iaf File1 File2 File3 File4
```

To add a suffix to a single filename, use the UNIX `mv` or `cp` command.

Converting one file

In a UNIX window, type:

```
filter [option1 option2 ...] infilename.suffix outfilename.suffix
```

where:

- *filter* is the name of the filter (for example, `wptomif`).
- *[option1 option 2 ...]* are command-line options that you can use with the filter. Available options are explained in the appropriate chapter of this chapter.
- *infilename.suffix* is the name of the file you want to convert (for example, `file.wp`).
- *outfilename.suffix* is the name of the converted file (for example, `file.mif`).

Using filterbatch to convert several files

Use `filterbatch` to convert a group of files using the same filter. You tell `filterbatch` which filter to use. In a UNIX window, type:

```
filterbatch [-d output_dir] filter [option1 option2 ...] infile1 infile2...
```

where:

- *output_dir* is the name of the directory where you want to save converted files.
- *filter* is the name of the filter.
- *[option1 option2 ...]* are command-line options that you can use with the filter. Available options are explained in the appropriate chapter of this manual.
- *infile1 infile2...* are the names of the files you want to convert.

For example, this command:

```
filterbatch wptomif Yourfile.wp Myfile.wp Hisfile.wp Herfile.wp
```

converts four WordPerfect files to MIF and identifies them with the suffix `.mif`. The `filterbatch` program automatically assigns a suffix to the files it creates.

For information about the filter options, see the `$FMHOME/fmunit/filterlist` file. The `AW4w` filters require a number when run from the command line. For example, the filterlist contains:

```
<'0490' AW4W MSWord MIF TextImport msw 'Microsoft Word Win 6.0/7.0'>
```

to run the Word 6 filter from the command line:

```
$FMHOME/bin/w4wtomif 0490 file.msw file.mif
```

dcatomif filter

The `dcatomif` filter converts documents from IBM Document Content Architecture Revisable-Form-Text (DCA) 1.0 and 1.1 formats to MIF.

Transferring DCA files

Before converting DCA files to MIF, you must make them accessible to the workstation on which you will be using FrameMaker products.

Customizing font translations

The `dca.am` file in the `filters` directory tells the filter which translation table to use for each font being converted to MIF. The `dca.am` file uses default translation values that are usually sufficient for most file conversions. If you need to change these default values, you must modify the `dca.am` file.

Caution: *If you're not familiar with MIF files or program code, you probably won't be comfortable modifying the `dca.am` file. In this case, your system administrator should customize the font translation tables.*

You have to obtain the character set definition for the new Graphic Font IDs (GFIDs) from your word processor's documentation or from IBM. The `dca.am` filter translates GFIDs 6, 11, 18, and 26 (Times, Times italic, Courier, and Courier italic, respectively) according to Appendix E of the IBM *Document Content Architecture: Revisable-Form-Text (DCARFT) Reference Manual*. GFID 26 is not optional; it must be defined in the `dca.am` file.

What the `dca.am` file looks like

Each line of the `dca.am` file describes a property or properties of the source font, followed by the corresponding information for FrameMaker products. The statements look like this:

```
<SrcChar <char_attributes_src>>
<MakerChar <char_attribs_frame>>
```

where `char_attributes_src` are the font characteristics in the DCA file and `char_attribs_frame` are the font characteristics in the MIF file.

For example, the following statements convert the DCA Times font (GFID 26) to Times, plain-style, 12-point characters:

```
<SrcChar <GFID 26>>
```



```
<MakerChar <FFamily Times> <FSize 12> <FPlain yes>>
```

The `dca.am` file that comes with your filter contains the following default values, which you can change.

Font property	MIF default
<FFamily string>	No default
<GFID n>	0
<FSize n>	0
<FPlain yes no >	yes
<FBold yes no>	no
<FItalic yes no>	no
<Code n>	No default
<Code C-hex>	No default
<Code C-octal>	No default
<Code C-char-const>	No default

Important: In all statements, *yes* and *no* responses must be in lowercase letters. All other words in the syntax must be capitalized as you see them in the list above. Font names may or may not have an initial cap, depending on how they are capitalized in the original application and FrameMaker product formats.

Inheritance and character substitutions

Once you specify a font property in the `<SrcChar>` statement or the `<MakerChar>` statement, the filter inherits and uses that property until it encounters another statement that changes it. This inheritance helps to keep the `.am` file short, because you don't have to repeat the conversion properties for each font and size. In addition to translating fonts, `dcatomif` can substitute characters if the FrameMaker product font and the font being translated do not share the same characters. (For a complete list of the FrameMaker character set, see the online manual *FrameMaker Character Sets*.)

The previous example illustrated a font translation. The example below includes a character substitution:

```
<SrcChar <Code 0x42>> <MakerChar <Code 0x89>> <Comment a circumflex>
```

Using the `<CopyFont>` statement

Rather than repeat all the properties for each font family, you can copy the information with a `<CopyFont>` statement. This saves time and effort and uses less memory in the filter program.

`<CopyFont>` tells the filter to use the translation table currently in use (including any changes in the details of the translation that were specified) to convert the font specified in `<SrcChar>`. The syntax is:

```
<SrcChar property> <CopyFont property>
```

For example, these statements:

```
<SrcChar <GFID 11> >
<CopyFont <GFID 26> >
```

tell the filter to apply the current version of the translation table for the Times font (GFID 26) to the Courier font (GFID 11).

When you use the `<CopyFont>` statement, note any exceptions to the default values with these substatements:

```
<PreChar property> and <PostChar property>
```

The `<PreChar>` property is the default value; the `<PostChar>` property is the change you want to make to that property. Enter these substatements before the `<CopyFont>` statement. For example, these statements:

```
<SrcChar <GFID 11> <Comment Courier 10>>
  <PreChar <FFamily Times>>
  <PostChar <FFamily Courier>>
<CopyFont <GFID 26> >
```

tell the filter to change the font family from Times to Courier before copying the font information.

Note: Unlike the `<SrcChar>` and `<MakerChar>` statements, `<PreChar>` and `<PostChar>` statements change only one property at a time.

Converting DCA documents to MIF

Always make backup copies of your files before converting them.

If you want to use a UNIX window to convert a DCA file to a MIF file, type:

```
dcatomif [option1 option2...] infilename.suffix outfile.suffix
```

where `[option1 option2 ...]` are command-line options that you can use with the filter, `infilename.suffix` is the name of the file you want to convert, and `outfile.suffix` is the name of the converted file.

For example, to convert the file `report.dca` to MIF, type:

```
dcatomif report.dca report.mif
```

Command-line options

The following table explains the command-line options for the DCA filter.

Use this option	To
-help, -h	Display <code>dcatomif</code> command syntax.
-V	Display the release number of the filter without running the filter

iaftomif filter

The `iaftomif` filter converts documents from Interleaf ASCII Format (IAF) release 8.0 (Interleaf 5) and IAF 6.4 (Interleaf 4) to MIF.

Preparing IAF files for conversion

In most cases, special preparation is unnecessary to convert IAF files. In rare cases where comments have been manually inserted into an IAF file, you should remove comments delimited by double hyphens that appear in a declaration or command. Otherwise, the filter might give unpredictable results. For example, edit the following declaration:

```
<!Class, Company,--this name stands for SuperVideo Stores, Inc.-->
```

to be:

```
<!Class, Company>
```

Customizing font translations

The `iaftomif` filter uses the file `filters/iaf.am` to build translation tables to convert fonts that do not translate directly. For example, Interleaf uses a proprietary font called Typewriter that `iaftomif` maps to a font family called Courier.

The `iaf.am` file tells the filter which translation table to use for each font being converted to MIF. The `iaf.am` file uses default translation values that are usually sufficient for most file conversions. If you need to change these default values, you must modify the `iaf.am` file. FrameMaker products support the fonts in the Interleaf 5 release. If you purchased additional fonts from Interleaf, you need to add support for the fonts by editing the `iaf.am` file.

Interleaf lets you use nonintegral font sizes (for example, 9.5 point). To convert Interleaf nonintegral font sizes, you must add conversion information to the `iaf.am` file. The default `iaf.am` file does not contain conversion information for nonintegral font sizes.

If the `iaf.am` file does not contain conversion information for a font, the filter converts the text to Times. If it cannot convert a type size, it converts text to 12-point. Any time the filter substitutes a different font, it displays a message.

This section describes the syntax of the `iaf.am` file and explains how to add or change font translations to get the effects you want in your documents.

What the `iaf.am` file looks like

Each line of the `iaf.am` file describes properties of the source character font, followed by the corresponding information for FrameMaker products. The statements look like this:

```
<SrcChar <char_attributes_src>>  
<MakerChar <char_attributs_frame>>
```

where `char_attributes_src` are the font characteristics in the IAF file and `char_attributs_frame` are the font characteristics in the MIF file.

For example, the following statements convert Typewriter font, plain-style, 10-point characters to Courier, plain-style, 9-point characters:

```
<SrcChar <FFamily Typewriter> <FPlain yes> <FSize 10>>  
<MakerChar <FFamily Courier> <FPlain yes> <FSize 9>>
```

You can change the following default values in the `iaf.am` file.

Font property	MIF default
<FFamily string>	No default
<FSize n>	0
<FPlain yes no >	yes
<FBold yes no>	no
<FItalic yes no>	no
<Code n>	No default
<Code C-hex>	No default
<Code C-octal>	No default
<Code C-char-const>	No default

Important: In all statements, *yes* and *no* responses must be in lowercase letters. All other words in the syntax must be capitalized as you see them in the list above. Font names may or may not have an initial capital letter, depending on how they are capitalized in the original application and FrameMaker formats.

Inheritance and character substitutions

Once you specify a font property in the <SrcChar> statement or the <MakerChar> statement, the filter inherits and uses that property until it encounters another statement that changes it. This inheritance helps to keep the `iaf.am` file short, because you don't have to repeat the conversion properties for each font and size.

In addition to translating fonts, the filter can substitute characters if the FrameMaker font and the font being translated do not share the same characters. (For a complete list of the FrameMaker character set, see the online manual *FrameMaker Character Sets*.)

The previous example illustrated a font translation from Typewriter to Courier. The example below includes a character substitution:

```
<SrcChar <FFamily Typewriter> <FPlain yes> <FSize 10>>
<MakerChar <FFamily Courier> <FPlain yes> <FSize 9>>
<SrcChar <Code 0x51>>
<MakerChar <Code 0x25>>
```

- The first line specifies the source character font (Typewriter), style (plain), and size (10) to be converted.
- The second line specifies the FrameMaker character font (Courier), style (plain), and point size (9) into which the source font is converted.
- The third line specifies the character that needs a substitution. In the Typewriter font, the `c/o` symbol is code 51. However, FrameMaker products have no `c/o` symbol.
- The last line indicates the FrameMaker character that takes the place of the `c/o` symbol in the translation. The hexadecimal code (`0x25`) is the FrameMaker character for a percent sign (%).

The following statements make additional changes to the same font translation:

```
<Comment This is the mapping for paragraph symbol>
<SrcChar <Code 0x57>>
<MakerChar <FFamily Times> <Code 0xa6>>
```

The source character font is still Typewriter, but the FrameMaker character font is no longer Courier; it is now Times. The paragraph symbol, code 57 in Typewriter, changes to code a6 in Times. The first line of code is a comment line, noting the character substitution.

In the following statements, the source character font is still Typewriter. The FrameMaker character font is now Symbol. The copyright character changes from code 45 in Typewriter to code d3 in Symbol.

```
<Comment circle Copyright>
<SrcChar <Code 0x45>>
<MakerChar <FFamily Symbol> <Code 0xd3>>
```

<CopyFont>

Rather than repeat all the properties for each font family, you can copy the information with a <CopyFont> statement. This saves time and effort, and uses less memory in the filter program.

<CopyFont> tells the filter to use the translation table currently in use (including any changes in the details of the translation that were specified) to convert the font specified in <SrcChar>. The syntax is:

```
<SrcChar property> <CopyFont property>
```

For example, these statements:

```
<SrcChar <FFamily Helvetica> <FSize 8>>
<CopyFont <FFamily Garamond> <FSize 12>>
```

tell the filter to apply the current version of the translation table for 12-point Garamond to 8-point Helvetica.

When you use the <CopyFont> statement, note any exceptions to the default values with these substatements:

```
<PreChar property> <PostChar property>
```

The <PreChar> property is the default value; the <PostChar> property is the change you want to make to that property. Enter these substatements before the <CopyFont> statement. For example, these statements:

```
<PreChar <FBold no>> <PostChar <FBold yes>>
```

tell the filter to change the default font weight to bold (FBold no to FBold yes) before applying it.

Note: Unlike the <SrcChar> and <MakerChar> statements, <PreChar> and <PostChar> statements change only one property at a time.

A more complex version of the preceding sample translation might read:

```

<SrcChar <FFamily Helvetica> <FSize 8>>
<PreChar <FBold no>> <PostChar <FBold yes>>
<PreChar <FItalic no>> <PostChar <FItalic yes>>
<CopyFont <FFamily Garamond> <FSize 12>>

```

These statements tell the filter to apply the translation table for 12-point Garamond to 8-point Helvetica, but, before applying it, to change the type style to bold italic.

An example

To add Helvetica Plain 20 to the `iaf.am` file:

- 1 Open the file with a text editor.
- 2 Search for Helvetica Plain.

You see:

```

<Comment ----->
<Comment Helvetica Plain.>

<SrcChar <FFamily Helvetica> <FPlain yes> <FSize 6>
<MakerChar <FFamily Helvetica> <FPlain yes> <FSize 6>>

<SrcChar <FFamily Helvetica> <FPlain yes> <FSize 8>
<PreChar <FSize 6>>
<PostChar <FSize 8>>
<CopyFont <FFamily Helvetica> <FPlain yes> <FSize 6>>

```

The first set of statements translate Interleaf Helvetica Plain 6-point type to FrameMaker Helvetica Plain 6-point type. The next set uses `<PreChar>` and `<PostChar>` statements to translate Interleaf Helvetica Plain 8-point type to FrameMaker Helvetica Plain 8-point type. You will see similar statements specifying the translation of Helvetica Plain at several type sizes. There is no translation, however, for Helvetica Plain 20-point type.

- 3 Locate the four statements that define the font translation for Helvetica Plain 18-point type:

```

<SrcChar <FFamily Helvetica> <FPlain yes> <FSize 18>
<PreChar <FSize 6>>
<PostChar <FSize 18>>
<CopyFont <FFamily Helvetica> <FPlain yes> <FSize 6>>

```

- 4 Copy these four lines so that the copied lines follow immediately after the originals.
- 5 In the `<PostChar>` statement in the duplicate set, change 18 to 20.

The eight lines should now read:

```
<SrcChar <FFamily Helvetica> <FPlain yes> <FSize 18>
<PreChar <FSize 6>>
<PostChar <FSize 18>>
<CopyFont <FFamily Helvetica> <FPlain yes> <FSize 6>>

<SrcChar <FFamily Helvetica> <FPlain yes> <FSize 20>
<PreChar <FSize 6>>
<PostChar <FSize 20>>
<CopyFont <FFamily Helvetica> <FPlain yes> <FSize 6>>
```

Converting IAF documents to MIF

If you want to use a UNIX window to convert an IAF file to a MIF file, type:

```
iaftomif [option1 option2 ...] infilename.suffix outfile.suffix
```

where *[option1 option 2 ...]* are command-line options that you can use with the filter, *infilename.suffix* is the name of the file you want to convert, and *outfile.suffix* is the name of the converted file.

For example, to convert the file `report.iaf` to MIF, type:

```
iaftomif report.iaf report.mif
```

Command-line options

You can use several command-line options with `iaftomif`.

Use this option	To
-help, -h	See the syntax of command-line options
-s	Convert a multicolumn IAF file to a single-column MIF file
-v	Set the filter to verbose to see messages while the filter converts a file
-V	Display the release number of the filter without running the filter

trofftomif filter

The `trofftomif` filter converts documents from troff format to MIF.

Preparing troff files for conversion

To prepare your files for conversion:

- 1 Do not run the preprocessors for `tbl` and `eqn` formats.

For instructions on handling those items after the conversion, see “Tables” on page 41 and “Equations” on page 41.

- 2 If they aren't there already, copy your files to a file system accessible to the computer on which you will be using the FrameMaker product.
- 3 If your troff file uses the `.so` command to produce header (include) files, use the UNIX `soelim` command to produce a master file that doesn't use header files.

Renaming troff files

Rename the files to use one of the following filename suffixes. The filename suffix determines how the file is filtered.

If a troff file	Use this filename suffix
Doesn't use a troff macro package	.m
Uses the -man macro package	-man (For other standard -man suffixes, such as .man, .1, .2, or .8, see the \$FMHOME/fmunit/fmimport file for instructions.)
Uses the -ms macro package	.ms
Uses the -me macro package	.me

Converting troff documents to MIF

If you want to use a UNIX window to convert a troff file to a MIF file, type:

```
trofftomif [-macropackage] infilename.suffix outfile.suffix
```

where `[-macropackage]` is the name of the macro package for the file (`-man`, `-ms`, or `-me`), `infilename.suffix` is the name of the file you want to convert, and `outfile.suffix` is the name of the converted file.

For example to convert the file `report.me` to MIF using the `-me` macros, type:

```
trofftomif -me report.me report.mif
```

Command-line options

The following table explains the `trofftomif` command-line options.

Use this option	To
-help, -h	See the syntax of command-line options
-V	Display the release number of the filter without running the filter

Touching up your documents

Your documents lose some formatting information in the troff-to-MIF conversion. `trofftomif` preserves as much formatting as possible but ignores these troff formatting instructions:

- Tables created in `tbl` format
- `eqn` codes
- Graphics in `pic` format

- Conditionals, nonconstant expressions, number registers, and preprocessor commands
- Document-specific definitions of macros, diversions, or traps

Tables

Tables in the troff tbl format aren't converted by `trofftomif`. For best results, don't run the preprocessor before running `trofftomif`. The filter moves the table information and the tbl codes to the FrameMaker product file. Delete the tbl codes and then re-create the table in the FrameMaker product by using the Convert to Table command on the Table menu. For more information, see the *FrameMaker User Guide*.

Equations

The `trofftomif` filter doesn't convert eqn codes; it passes the codes to the FrameMaker product file as text. You have to re-create the equations in the FrameMaker product. For information, see your user's manual.

Paragraph formats

After converting a few troff files, you may find that `trofftomif` inserts forced returns after more paragraphs than you like. If the paragraph formats in the converted file need adjusting, use the Paragraphs>Designer command on the Format menu.

troff commands

The `trofftomif` filter converts most troff requests, troff escape sequences, and macro references. It sends any unconverted commands to the FrameMaker document as text. Touch up your document as follows:

- 1 Use the Find/Change command to locate any troff commands that weren't converted.
- 2 Look at the surrounding text and determine what you need to touch up.
- 3 Delete the troff commands.

Graphics

Your troff files may contain pic graphics, which aren't converted by `trofftomif`. Use the Tools palette to re-create the graphics.

fmimagetoimage filter

The `fmimagetoimage` filter converts an imported image or inset stored in a FrameImage format in a MIF file to the Sun raster file (rf) image format. The filter converts color or black-and-white images imported by FrameMaker products. For more information about the FrameImage format, see the online manual *MIF Reference*.

Preparing to convert Framelimage files

The `fmimageimage` filter converts only the first graphic in a FrameMaker document; subsequent graphics are ignored. To convert several graphics within the same FrameMaker document, copy each graphic and paste it into its own FrameMaker document.

Converting Framelimage files

- 1 Open the FrameMaker document containing the graphic.
- 2 Choose Save As from the File menu, enter a filename for the graphic, and choose Interchange (MIF) from the Format pop-up menu.
- 3 Type the following command in a UNIX window:

```
fmimageimage [option1 option2 ...] infilename.suffix outfilename.suffix
```

where `[option1 option 2 ...]` are command-line options that you can use with the filter, `infilename.suffix` is the name of the MIF file you want to convert, and `outfilename.suffix` is the name of the converted file.

Command-line options

The following table shows `fmimageimage` command-line options.

Use this option	To
<code>-h</code> or <code>-help</code>	See a description of the command-line options
<code>-v</code>	Display the release number of the filter without running the filter