

Converting Between Word and Frame

Introduction

Many writers break into a sweat at the thought of converting large files between Word and FrameMaker. It turns out, however, that with some preparation, the process can nearly fly in *either* direction.

Most of the rules and guidelines for painless conversions apply to either situation (Word > Frame or Frame > Word); to that end, I tried to write as much of this as I could using generic terms like *source* (the file being converted) and *target* (what the file is being converted to). Also, while this guide is indeed a “how-to” on “how to make conversion easier,” it is *not* a how-to on how to perform individual tasks. For example, this guide will tell you to remove Word’s automatic numbering, but it will not tell you how to actually go about doing it. It is your responsibility to be fluent enough in Frame and Word to perform the necessary tasks.

The process described herein is valid for Word 97 and 2000 (differences noted when I’m aware of them), and FrameMaker 5 and 6, running under Windows 98 Second Edition. I have not done a lot of conversions on Macintosh, but from the few I’ve done I’ve found the processes are the same.

When I think of new things to add I will post a new document at the web site, so check back once in a while and see if it’s been updated. I’m open to suggestions, too. Finally, do you remember the robot Ash from the movie Alien? Remember when his severed head said, “you have my sympathies?” Turns out they asked him about doing serious documents in Word.

Tim Murray
TechKnowledge Corp.

www.TechKnowledgeCorp.com
TMurray@TechKnowledgeCorp.com

A summary

It’s the dream of many that one should be able to convert between these programs just as is. They think, “if the filters were any good, I could just convert from one to the other straight away.” In a perfect world, that would be the case, but it’s not a perfect world, so you’re going to have to massage your document to get it ready for the ordeal. This causes a lot of concern for many people because they think that the task of preparing the source for the conversion is destroying it. Well, it is, in a way, but you’re leaving it to the target to put it back together.

It turns out that if you really know Frame and Word, the following sentence may be all you need and you don’t even have to read the rest of this document: The goal is to get your source document to look nearly like a plain-text document yet retain its style names, which will be matched up with target styles or tags later, and letting the target application do the things it does best. That’s it. Really. If you need more info, read on. . . .

File formats

In the direction of Frame > Word, some find they like the “save as Microsoft RTF” or “save as Japanese RTF” approach, but if the file is prepared properly, I’ll bet my last dollar that saving as a Word document will provide better results. If the save takes all night or crashes, then, in my experience, the file wasn’t prepared properly. The same is true for Word > Frame: you could save your Word file as an RTF and then import that, but I think that a regular .doc format is better.

Therefore, this entire document assumes that if you’re converting from Frame to Word, you will use a “save as Microsoft Word”; if you’re converting from Word to Frame, then you’re either doing an open or an import in Frame of a regular Word .doc file.

General preparation of the source file

Anything in this section is universal for either direction.

1 Manage your styles and tags.

As often as possible, use like-named styles in both applications. In fact, with the exception of matching Word’s default Normal with Frame’s default Body, there is little reason to have disparity in style names. See also “The target template” on page 8.

The less styles or tags a converted file has, the easier it will be to work with, so you will also want to remove styles or tags that are unused and merge those that are very similar. Later, if you find that you need extra styles or tags, you can add them.

2 Kill the table of contents.

Because Frame does such an excellent job of TOCs, there is no reason at all to keep anything in Word having to do with a TOC, whether the TOC is a separate document or a section of a file. If you’ve use Word bookmarks to generate a TOC, delete them all. And since Word sucks so bad at TOCs and it will be necessary to do it later anyway, don’t even bother trying to make a Frame TOC into something Word can use; just make a fresh TOC in Word.

3 Remove headers and footers.

There is little hope of having them appear in the target as you want anyway, so just plan on doing them from scratch. Both programs do things differently in terms of automatic contents.

4 Remove local paragraph overrides.

Hopefully, local overrides were done only occasionally.

- **Word:** Select the entire document and press Ctrl+Q.
- **Frame:** One way is to use Import > Formats and take the option for removing overrides. Another way is to select a few key paragraphs (one at a time, of course), open the Paragraph Designer, and do an update all, choosing remove overrides whenever it appears.

5 Remove local character overrides (optional).

You may want to test this first, as I have found some oddities that come and go. For example, say a certain paragraph is entirely roman except for the 15th word, which is italic. If the italic was set using a character style, the target might assign that style to the 15th word and to the remaining words of the paragraph. Therefore, I suggest you leave them in at first and see how it goes—if your results suck, then take them out of the source and add them back to the target.

- **Word:** Select the entire document and press Ctrl+Space.
- **Frame:** First, I'll tell you what *not* to do: Do *not* select the entire document and click Default ¶ Font! Try it sometime on a test document and see what happens. (This is one Word command I wish Frame had.)

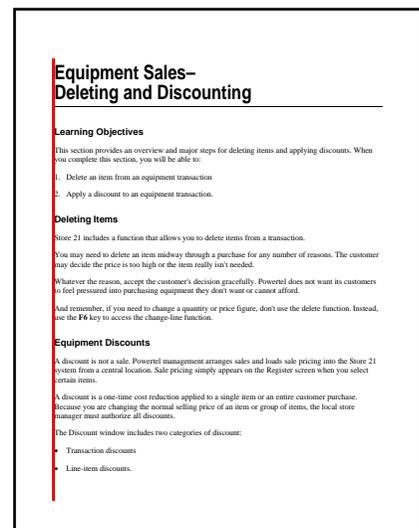
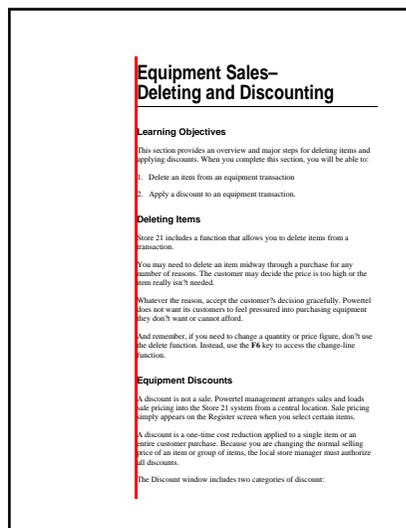
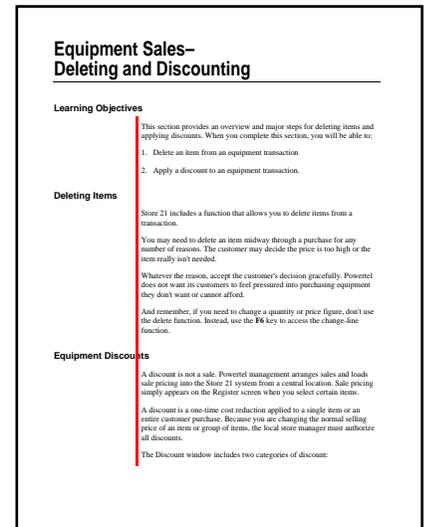
If you only delete any character tags from the library it won't remove them from the text so you'll have to remove them from a MIF.

6 Set margins to zero.

Consider the page shown to the right.

This could be a Frame or a Word document, and each program has its own way of creating this kind of layout.

Below I show the goal, which is to get everything “lined up” along a uniform left margin. Whether or not the source ends up looking like the figure on the left or the right really doesn't matter: the point is that the left edge of the flow is the ruler's zero:



- **Word.** If the author used negative indents for headings and the body is at zero, make the headings zero (results will look like left image). If positive indents were used for the body, then make the body zero (right).
- **Frame.** Frame doesn't do negative indents, so chances are the headings were done using side heads. Make all paragraphs (column-spanning, side-head, etc.) into regular in-column paragraphs, then change your master pages to not use any side head at all. (See also the next item.) Do the paragraph changes before the master page change because any you missed will be easier to catch.

7 Remove things the target program can't interpret.

- **Word:** Remove borders, WordArt, callout boxes, arrows, text frames, etc. Text frames are commonly used to do what Frame would call a side head. If you want to retain the text, you will need to copy it into the regular flow.
- **Frame:** Remove frames above/below and Frame-drawn art. Convert variables to text (you'll get the converted text anyway, but I just like to do this because it's less work on the filter). For cross-references, see "Cross-references" on page 6. Convert all types of anchored frames to either "at insertion point" or "below current line".

8 Save the source file.

When you save your cleaned source file . . .

- **Word:** Make sure the option Allow Fast Saves is turned off, or, better yet, save the file as a totally new file (but generally speaking, Allow Fast Saves should be turned off anyway).
- **Frame:** Remember to place a .doc extension on the saved file, or else you'll blow away your .fm file.

Master pages and sections

Although master pages (Frame) and sections (Word) are not the same constructs, they are frequently used for the same purpose: To impart a page layout that is in some way different from the rest of the document.

Word to Frame

In Word, sections are frequently used to insert changes in headers and footers, such as when a header reflects the chapter name. Since you'll be doing things differently in Frame anyway (as in breaking chapters into files), remove all section breaks (search for ^b).

Frame to Word

Apply the regular L/R master pages to the whole file, then remove the custom masters. (Technically you don't have to remove the masters, but I just do it anyway because that's the way I am.)

Headings in general

Word to Frame

Frame could care less if a heading is named Heading 1 or Fred_Flintstone, so you don't really have to mess with Word's style names before the conversion unless you want to change the style names to match the target template.

Frame to Word

You probably know that Word gives special treatment to paragraphs with certain names, particularly the headings of Heading *n*. In other words, Word can deal with Heading 1 much differently than it can Fred_Flintstone. Your knowledge of Word and the kind of numbering or cross-referencing you expect to do later will determine if you want to change Frame's heading names before the conversion. Of course, you can do a find-and-replace later.

Numbering

Word to Frame

If you come across a Word document that uses a great deal of automatic numbering *and* where the numbering is based on styles, that is a good thing: Simply blow away the numbers (but not the styles) in Word, and don't worry if things look like crap after you do. Later, when Frame opens the file, you can attach a template with numbering applied and find your numbering is pretty much done for you. You will probably have to do some resetting to 1 and so on, but that's cake in Frame.

In contrast, you might leave the numbers in a Word document that's been poorly done. That is because later, in Frame, the bullets and numbers will give you something to find ([0-9], for example) and change (paste an autonumbered tag).

Frame to Word

I'm not going to even bother with this section. The reason is that so much depends on how you'll do it in Word. You might use SEQ fields, autonumbers assigned to headings, or simply do the numbering yourself. I will simply leave it to you to figure out the best way to do it.

If you want to experiment, first give a shot to leaving the numbering in the Frame document. If it helps when it gets to Word, then great. If it doesn't or it makes things worse, then take it out and put it back in later, in Word.

Bullets

Word to Frame

Like numbering, bullets were hopefully done with styles and not manual overrides. If styles were used, then keep the style names but remove the bullet from the definition. When the file gets put into Frame, the bullets themselves should appear as a result of Frame's numbering.

If the bullets were done using overrides or were typed in by hand, you might find your imported paragraphs contain two bullets—one that’s defined by the Frame tag, and one that was simply picked up during the import. If you have extras, copy a good bullet tag to the clipboard, then search for a bullet followed by a tab and replace with a paste. Follow up with another find & change and delete the extra bullets and tabs.

Frame to Word

Word tends to maintain bullets pretty well. You may have to mess with the default font and size of the bullet, but in many cases, you should find the Word style is defined as you want.

Cross-references

If you know Word, then you know that it can only make a cross-reference to something in the same file—it can’t look inside another file like Frame can.

Word to Frame

When Frame reads a Word file with cross-references, they are usually maintained. However, what you get is a rather odd marker like “_Ref473865239”. You could leave it as that and get the job done, but maintenance in the long run could be a chore. Therefore, you may want to strip them out of Word before the conversion and redo them later.

Frame to Word

What you get depends on the reference. A Frame reference like “See Cross-references on page 5” might end up in Word as “See Cross-references on page 1” with an odd character style applied to the heading name. I’m not sure why this happens, but it seems that Word will update a change to the heading name correctly but the page number is almost always incorrect. For example, if you change the heading (let’s say it’s really on page 5) to from “Cross-references” to “Making Cross-refs”, Word will update its cross-reference to “See Making Cross-refs on page 2”. Where the heck it gets page 2 is beyond me, except that the page number seems to be the page where the cross-reference resides, not the bookmark.

The fact that an odd character style is applied could be used to your advantage: It will give you something to look for on screen as you page through the Word version.

Another oddity I’ve found is that when the Frame-exported file is opened in Word, you find that a section break has been inserted just before the actual reference; for example, you might end up with something like “See [section break] Cross-references on page 1”.

Indexes

I have found that both programs can read the other program’s index markers okay. I have not really done a lot of conversions of indexes with multiple levels, so that’s about all I can say.

Except for this: I received a report that when converting from Frame to Word, the presence of an apostrophe in a marker could crash either Frame's save as or Word when it opened the doc file (I forget which). I tested this using Frame 6 to Word 2000 and did not crash, but be aware of this.

Artwork

Graphics in this section refer to imported or referenced artwork, not art drawn with the tools of the source application.

Word to Frame

Graphics that were pasted into Word will appear in Frame. Imported graphics are a different matter, because the link and save settings affect what happens on the Frame side. The following table lists what you will get in Frame as the result of the various combinations of Word's linking and embedding.

Linked	Inserted (Saved)	General Appearance and Anchor Result	Link Maintained?
✓		Forced into 1-inch square frame. Anchored at insertion point.	Yes
	✓	Full size. Anchored at insertion point.	not applicable
✓	✓	Full size. Anchored at insertion point.	No

Of these results, my favorites is the option that maintains the links. The reason is that you can select a picture, do a File > Import > File, and find that Frame will already have the graphic's file name in the dialog box. Just press Enter, and bang, a new graphic is imported.

Frame to Word

A Frame file exported to Word will maintain graphics whether the graphics are linked or embedded; however, Word will have no knowledge of the link. If you plan on using linked graphics in Word (as you should), then consider removing the graphics prior to the export: it greatly reduces complexity, and you gotta put them back anyway. If you choose to remove the graphics, make sure to remove the enclosing frame, not just the graphic that's inside the frame.

Another tip is to record the files you used. Here is a technique I use for linked graphics:

1. Select a linked picture.
2. Choose File > Import > File. When the dialog box appears, you'll see that the file name is highlighted. Don't press Enter.
3. Copy. Yes, you can copy text in a dialog box field to the clipboard.

4. Close the dialog box, return to the document, and paste the text next to the figure or in its own paragraph immediately above or below the old location.
5. Delete the figure's frame.

What you've done is removed the graphic but kept the file name on the page. Now, when you begin doing new graphic imports, you'll save a ton of time because you have the graphic file name right there on the page.

It also turns out that when you go from Frame 6 to Word 97, all graphics, whether linked or embedded, turn out as floating graphics. That, in and of itself, might be reason enough to kill the graphics before the conversion. From Frame 6 to Word 2000, they (thankfully) end up as in-line graphics.

Tables

Word to Frame

“Unspan” spanned columns. A spanned column can come into Frame about one pica wide and thus screw things up. Just remove the spans, then add them back in later.

A note about heading rows. Oddly, Frame does not detect the presence of a heading row in Word. One solution is to use Word (no, seriously) to process the Frame MIF. At the end of this document is text you can use as a Word macro to convert the top row of all tables into a header row. But note that if it encounters a table that already has a header, it will take the first nonheader row and make a header row out of it, so you'll have two.

Frame to Word

I've found tables in the direction of Frame to Word work a bit better than Word to Frame; about all you'll have to mess with are spanned cells: From Frame 5, Word sees horizontally spanned cells not vertically spanned; from Frame 6, Word sees both.

Books and master documents

Work on a file-by-file basis. Join them up later or make as book as needed. Never use Word's Master Document feature. It's junk, and it's even worse in Word 2000.

The target template

When you open or import the source file into the target application, you're going to want to get it looking good—close to the final layout—as soon as possible. The best way to do this (in my opinion) is to prepare a template that has the styles or tags you'll be using.

One question to answer is whether to use similar style or tag names or use a completely new set. Personally, I prefer the former because there is less manual labor, but this is a matter of personal preference.

You may want to assign the target's template to the target file both before and after you do these next final fixes.

Final Frame fixes

Word to Frame

At this point you've opened or imported your prepared Word document in Frame.

1 Remove and combine unwanted character and paragraph tags.

After you've opened a Word document in Frame, you may find that you've gained an unwanted character tag called Default Paragraph Font (Frame already has one called Default ¶ Font). When you remove a tag from Frame's catalog, that is indeed *all* you've accomplished—if it's been assigned to text, it's *still* assigned to text. The only way to actually remove it from text is to either select the text and choose Default ¶ Font (your goal nine times out of 10), assign it something else, or save the file as a MIF and delete all occurrences of the string "Default Paragraph Font" minus the quotes. The latter method is very fast.

Many times in Word documents I find the author has created numerous styles that look nearly identical, so you may want to make them all the same tag to start, and add any special tags later if you really need them. Combining tags is easy in Frame. First, let's assume you have three tags—T1, T2, and T3—and you want T2 and T3 to be named, and to look like, T1.

1. Make sure the target template has a T1 that's just as you like it and that does not have a T2 or T3.
2. In the file to be changed, click inside a T2 paragraph, then open the Paragraph Designer.
3. Change the name to T1. Don't use the pull-down arrow; instead, type right over it.
4. Choose Update All. When Frame asks if you want to change all T2 to T1, answer yes.
5. Do the same for T3: change it to T1.
6. Delete T2 and T3 from the catalog, if they are still lingering.
7. Open the file from step 1, and import its paragraph catalog. I would also suggest taking the option to remove overrides.

Experienced Framers know that you can get around bothering with the import. However, it's sort of a safety net for the new folks.

2 Remove custom table ruling and shading.

Even when the Word table does not have what appears to be custom settings, Frame can assign custom settings to the table. There is a way to remove these custom settings from the entire document in a flash.

1. Select the entire document.

2. Choose Table > Custom Ruling & Shading.
3. Select both of the “From Table” settings, and select all check boxes.
4. Click Apply.

The reason why this works is that selecting the table anchors is sufficient to select the tables.



Frame to Word

1 Remove section breaks.

Seems like Word loves section breaks . . . when it reads a doc file saved from Frame, it has a tendency to slap them down all over the place. Remove them globally then add them back in where you need them.

2 Combine similar tags.

In Word, you can use the find/replace. You find text of one format and replace with another.

3 Remove unwanted tags.

Word works a bit differently than Frame when you remove a style. When you delete a style, Word reassigns the Normal style to any paragraph that was assigned the style you just deleted. This can be very handy, however, in cases where the author had used a number of paragraph tags that were similar—close enough to at least start out as Normal, in any case.

If you don't want to reset them to Normals, use several finds (of the similar paragraph styles) and replaces (with style you want). Remove the style after you've changed them all.

FrameMaker Tools

There are some very useful (and sometimes free or inexpensive) tools out there to help with some of the Frame cleanup, both before and after the conversion.

Archive collects the files in a book and other files it references (graphics, for example), and makes a copy in a separate location. E-mail the fellow directly at bruce_foster@compuserve.com.

Remove Local Formatting strips local overrides from a file or from a selection, sort of like Word's Ctrl+Q and Ctrl+Space. **Clean Import** deletes all paragraph and character tag names, table definitions, cross-reference formats, conditional text tags, master pages, reference pages from the current document, then allows you to select formats from a selected template.

Source: <http://www.electropubs.com>

Hunt Overrides from Chris Despopoulos of CudSpan generates a view-only window with a list of paragraph overrides on body pages. Click an item in the list to jump to the corresponding paragraph. His **DoBatch** utility does some extra useful book-wide functions presently not provided by Frame.

Source: <http://www.telecable.es/personales/cud/>.

Toolbox is a collection of tools you can buy as a set or individually. The tools do a lot of functions, but the tool I find particularly handy is the format renamer. It lets you change a tag name throughout a book, or combine multiple tags into one. But note that it does not change the cross-references that use the tag.

Source: <http://www.systemec-gmbh.com>

Alert is a nifty little tool from Silicon Prairie. When you open a file, it looks for a variable named Alert. If it finds that variable, it displays its contents in a small Alert dialog—handy for reminders for a workgroup working with templates. It's just a dll that sits in your plug-ins folder. Also, **Paragraph Tools** provides additional capabilities for working with paragraphs in FrameMaker documents. Capabilities include listing paragraph tag usage, removing unused paragraph tags, converting paragraphs to use different tags, and locating and removing paragraph tag overrides.

Source: Silicon Prairie <http://www.kagi.com/SPSoftware>

The Word-to-Frame table macro

Run the procedure on an MIF file. It changes the first row of each imported table to a heading row. Note that if it encounters a table that already has a heading row, it takes the top regular row and adds it to the heading, giving you two heading rows. Courtesy Stephen Sealy.

```
Sub FixTableHeadings()  
'Enable the error-handling routine.  
On Error GoTo ErrorHandler  
  
'Go to the top of the file.  
Selection.HomeKey Unit:=wdStory  
  
'Change the first row of each "Format A" table to a heading row.  
Do While True  
'Find the table.  
Selection.Find.ClearFormatting  
With Selection.Find  
'Text = "<TblTag `Format A`>"  
.Text = "<TblTag `Format A`>"  
.Replacement.Text = ""  
.Forward = True  
.Wrap = wdFindStop  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute  
If Selection.Find.Found = False Then Exit Do  
  
'Find the first row.  
Selection.MoveDown Unit:=wdLine, Count:=1  
With Selection.Find  
'Text = " <TblBody ^p <Row "  
.Text = " <TblBody ^p <Row "  
.Replacement.Text = ""  
.Forward = True  
.Wrap = wdFindStop  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute  
If Selection.Find.Found = False Then Exit Do  
  
'Delete <TblBody marker.  
Selection.MoveLeft Unit:=wdCharacter, Count:=1  
Selection.EndKey Unit:=wdLine, Extend:=wdExtend  
Selection.MoveLeft Unit:=wdCharacter, Count:=1, Extend:=wdExtend  
Selection.Delete Unit:=wdCharacter, Count:=1  
  
'Insert the <TblH  
Selection.TypeText Text:=" <TblH "  
Selection.MoveDown Unit:=wdLine, Count:=2  
  
'Find the end of the first row.  
Selection.Find.ClearFormatting  
With Selection.Find  
'Text = "> # end of Row"  
.Text = "> # end of Row"  
.Replacement.Text = ""  
.Forward = True  
.Wrap = wdFindContinue  
.Format = False  
.MatchCase = False  
.MatchWholeWord = False  
.MatchWildcards = False  
.MatchSoundsLike = False  
.MatchAllWordForms = False  
End With  
Selection.Find.Execute  
  
'Insert the end tag for <TblH and the beginning tag for <TblBody.
```

```
Selection.MoveRight Unit:=wdCharacter, Count:=2
Selection.TypeParagraph
Selection.MoveUp Unit:=wdLine, Count:=1
Selection.TypeText Text:=" > # end of TblH"
Selection.TypeParagraph
Selection.TypeText Text:=" <TblBody "
Selection.MoveDown Unit:=wdLine, Count:=2

Loop

' Exit to avoid handler.
Exit Sub

' Error-handling routine.
ErrorHandler:
MsgBox Err.Description
End Sub
```